

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

Katedra měřicí a řídicí techniky

DIPLOMOVÁ PRÁCE

Vizualizační a informační systém pro teplotní monitoring vrtů.
Information System for Temperature Monitoring in Boreholes.

2011

Richard Vašut

Zadání diplomové práce

Student: **Bc. Richard Vašut**
Studijní program: N2649 Elektrotechnika
Studijní obor: 2601T004 Měřicí a řídicí technika
Téma: Vizualizační a informační systém pro teplotní monitoring vrtů.
Information System for Temperature Monitoring in Boreholes.

Zásady pro vypracování:

1. Rozbor současného stavu sběru dat z měřicího polygonu vrtů na VŠB-TU Ostrava.
2. Analýza možností přenosu měřených dat z malého a velkého měřicího polygonu na databázový server. Analýza možností internetové vizualizace těchto dat.
3. Návrh a realizace lokální vizualizační aplikace.
4. Návrh a realizace internetové vizualizační aplikace.
5. Testování vytvořených aplikací.
6. Zhodnocení výsledků.

Seznam doporučené odborné literatury:

1. KOZIOREK, Jiří. Programovatelné automaty a vizualizace řídicích systémů. Výukový text pro studenty oboru Měřicí a řídicí technika, VŠB – TU Ostrava, 2009.
2. Firemní technická dokumentace pro použité komponenty řídicího a vizualizačního

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Jiří Koziorek, Ph.D.**

Datum zadání:

Datum odevzdání:

doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení Studenta

„Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.“

V Ostravě

Podpis

Poděkování

Za cenné informace a připomínky k práci děkuji svému vedoucímu doc. Ing. Jiřímu Koziorkovi, Ph.D. Dále děkuji své rodině za podporu při studiu.

Abstrakt

Tato diplomová práce se zabývá tvorbou vizualizačního a informačního systému pro velký a malý výzkumný polygon. Na každém polygonu je několik vrtů, které jsou vybaveny z větší části teplotními čidly PT 1000. Dochází ke sběru dat z těchto měřicích systémů a za pomoci dalších prostředků jsou data archivována v databázi. Tato procedura prošla vlastním vývojem, kde byl použit vodopádový model. Z této databáze jsou následně data k dispozici v lokální a internetové vizualizaci, kde je možnost zvolení určitého vrtu společně s časovým rozmezím. Lokální vizualizace byla vytvořena ve SCADA systému od společnosti Promotic a internetová vizualizace byla vytvořena v prostředí ASP.NET. Analyzování těchto dat je možné nejen ve vizualizaci, ale i na vlastním počítači.

Abstract

This thesis deals with creating information system for large and small research polygons. There are several boreholes on both polygons, which are in majority equipped by PT 1000 temperature sensors. Data acquisition is performed and measured data is archived in the database. This software procedure is designed by waterfall model. Data from database is achievable in local and internet visualizations, where there is choice of boreholes selection with configurable time range view. The local visualization was created in Promotic SCADA system and internet visualization was created in ASP.NET framework. Data analysis is possible both in visualization and also on a personal computer.

Klíčová slova

Čidla, Databáze, Data logging, DataHub, FTP, ICPDAS, ODBC, OPC klient, OPC server, polygony, sběr dat, SCADA

Key Words

Data acquisition, database, Data logging, DataHub, FTP, ICPDAS, ODBC, OPC client, OPC server, polygons, SCADA, sensor

Seznam použitých symbolů a zkratek

ASP.NET	-	tvorba webových aplikací
BR	-	ButtonRadio
CB	-	ComboBox
COM	-	Component Object Model / technologie pro komunikaci mezi softwarovými komponentami
CSV	-	Comma-separated values / hodnoty oddělené čárkami
DCOM	-	Distributed Component Object Model / technologie pro komunikaci mezi softwarovými komponentami v rámci sítě
MP	-	Master Page/ hlavní stránka
MVP	-	Malý výzkumný polygon
MPI	-	Multi Point Interface / sběrnice určená pro průmysl
ODBC	-	Open Database Connectivity / standard pro přístup k databázovým systémům
OPC	-	OLE for programing kontrol / standard pro přenos dat v RT
PLC	-	Programmable logic controller / programovatelný logický automat
VVP	-	Velký výzkumný polygon
HTTP	-	Hypertext Transfer Protocol / protokol pro přenos hypertextových dokumentů
SMTP	-	Simple Mail Transfer Protocol / protokol pro přenos elektronické pošty
FTP	-	File Transfer Protocol / protokol pro přenos souborů
PHP	-	Hypertext Preprocessor / skriptovací programovací jazyk
MSSQL	-	Microsoft SQL Server / databáze
MySQL	-	database/databáze
RS232	-	Seriál port / sériové rozhraní
RT	-	real time / reálný čas
SCADA	-	Supervisory Control And Data Acquisition / řízení kontroly a sběr dat
SMDS	-	SiteMapDataSource/ zdroj dat hlavního menu
USB	-	Universal Seriál Bus/ univerzální sériový port
VxComm	-	Virtual Communications Ports / virtuální komunikační port

Obsah

1. Úvod	1
2. Rozbor současného stavu sběru dat z měřicího polygonu vrtů na VŠB-TU Ostrava	3
2.1 Malý výzkumný polygon MVP	4
2.2 Velký výzkumný polygon VVP	6
2.3 Účel a rozdíly sběru dat u obou polygonů	7
2.4 Popis čidla PT1000	7
3. Analýza možností přenosu měřených dat z malého a velkého měřicího polygonu na databázový server. Analýza možností internetové vizualizace těchto dat	10
3.1 Možnosti přenosu měřených dat do databáze	10
3.1.1 OPC standard a přístup do databáze	12
3.1.2 Jiné standardy s možností přístupu do databáze	15
3.1.3 Databáze	18
3.1.4 Databáze a zvolený způsob ukládání dat	21
3.2 Možnosti internetové vizualizace	22
3.3 Realizace aplikací	23
4. Návrh a realizace lokální vizualizační aplikace	24
4.1 Návrh	24
4.2 Realizace	24
5. Návrh a realizace internetové vizualizační aplikace.....	32
5.1 Návrh	32
5.2 Realizace	33
5.2.1 Programová část	33
5.2.1 Popis funkce webové vizualizace	40
6. Testování vytvořených aplikací.	43
7. Zhodnocení výsledků.	45

1. Úvod

V dnešní době je značně diskutovaným tématem využití obnovitelných zdrojů. Konkrétně hlubinné vrtý společně s tepelnými čerpadly mají určitý energetický potenciál. Nasazení této technologie dnes využívají k topení i malé rodinné domy. Hodně firem dnes nabízí tepelná čerpadla a zákazník si vybírá nejen podle ceny, ale taky podle parametrů. Tím nejhlavnějším parametrem je energetický zisk. Za tímto účelem se stále zlepšují nabízené produkty. Největší význam při zlepšování energetického přínosu má určitě výzkum.

Za účelem zkoumání bylo v areálu VŠB-TU Ostrava vybudováno několik takovýchto vrtů a jsou rozděleny na dva samostatné měřicí celky. Jedná se o malý výzkumný polygon (MVP) a velký výzkumný polygon (VVP). Na všech vrtech je zkoumán vliv teplotních změn horninového masivu při odebírání tepelné energie. Některé vrtý jsou přímo pro tepelná čerpadla a některé jsou monitorovací. Jejich umístění je okolo vrtů pro tepelná čerpadla. Tím se kontrolují i teplotní odezvy ve vzdálenějších místech. Z toho vyplývá, že sběr těchto informací a následné ukládání v čase je důležitým elementem celého systému.

Vrtý obsahují teplotní senzory a v každém vrtu je rozmístěno několik čidel v určitých hloubkách. Na MVP jsou teplotní údaje sbírané přes PLC automat firmy B&R. Na VVP jsou data sbíraná měřicím systémem zvaným ICPDAS. B&R ukládá svá data na FTP server ve formátu CSV. Dostupnost těchto dat je možná přes vzdálený přístup k FTP, kde jsou všechna data ukládána do měsíčních souborů. ICPDAS svá data ukládá do databáze Access a je zde vytvořena vizualizace, která zobrazuje aktuální i historické záznamy.

Cílem této diplomové práce je vytvořit vizualizační a informační systém na míru pro oba měřicí polygony, kde bude zajištěn dlouhodobý sběr dat ze všech vrtů společně s náhledem na tyto uložené informace.

Druhá kapitola rozebírá dosavadní stav na obou měřicích polygonech. Popisuje jednotlivé vrtý a různě zvolené prostředky pro sběr dat. Součástí kapitoly je popsání čidla PT 1000, jeho vlastností a způsobů zapojení.

Třetí kapitola se zaměřuje na možnosti přenosu měřených dat z malého a velkého výzkumného polygonu do databáze. Jako první je zmíněna problematika OPC, která je jako jediný standard u obou měřicích systémů. Bylo třeba zanalyzovat trh nabízeného softwaru, který poskytuje propojení OPC s databázemi. Následně je také rozebrán standard ODBC pro databáze. Mimo jiné kapitola představuje i další možnosti přenosu dat do databázových systémů bez použití OPC standardu, jako je např. použití Ethernetu a jeho protokolů. Je třeba se také zmínit o problematice potřebného strukturování a ukládání dat v databázi a v neposlední řadě jsou rozebrány možnosti internetové vizualizace.

Čtvrtá kapitola pojednává o návrhu a realizaci lokální vizualizace navržené v Promoticu. Mimo tuto vizualizaci je vytvořena ještě jedna vizualizace založená na protokolu DDE. U obou aplikací jsou popsány principy tvorby.

Pátá kapitola se zaměřuje na návrh a realizaci internetové vizualizace. V návrhu jsou rozebrány základní požadavky, které by vizualizace měla mít. Realizační část je rozdělena z hlediska programové a uživatelské části. Programová část popisuje principy praktického provedení. Uživatelská část slouží jako návod pro snadnou manipulaci a analýzu potřebných dat z internetu.

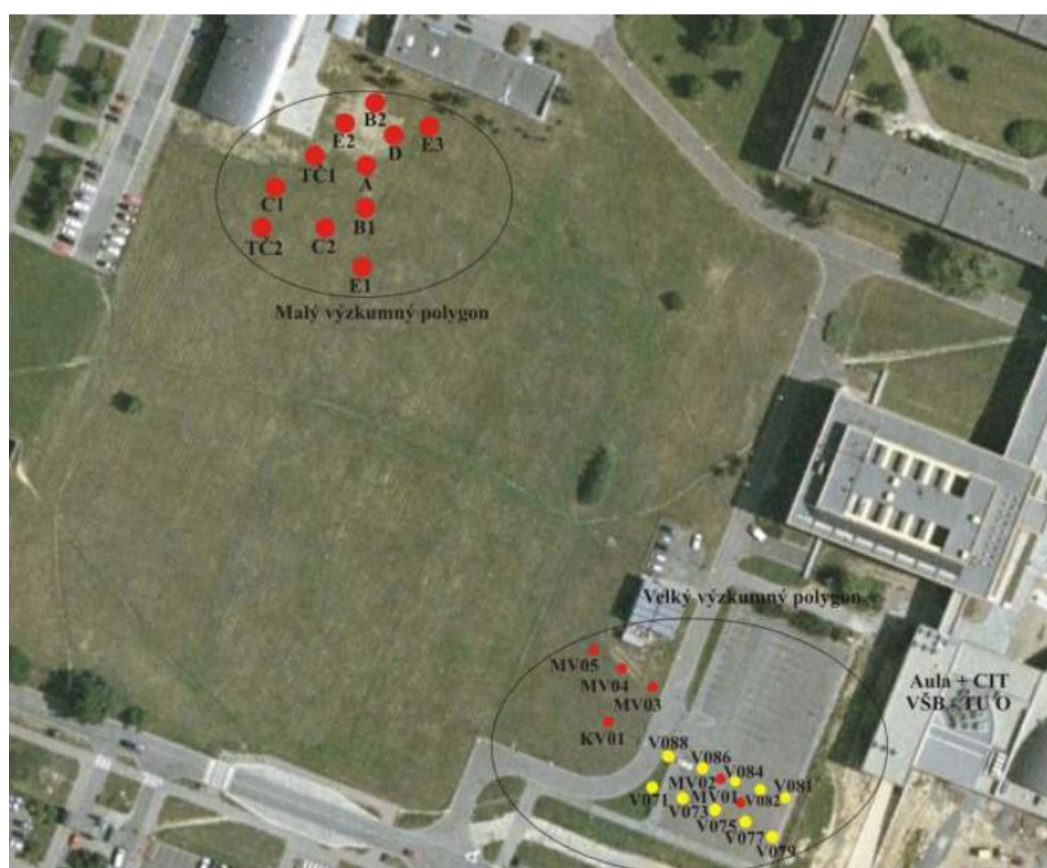
Neméně důležitá je poslední kapitola, která se zaměřuje na testování vytvořených aplikací. Jsou realizovány testy a zároveň je popsáno několik důležitých poznatků, které se týkají funkčnosti a bezpečnosti.

Protože celý tento informační systém zahrnuje funkci sedmi aplikací, které navzájem komunikují, jsou vytvořeny a následně umístěny návody pro spravování a nastavení dílčích programů v příloze práce.

2. Rozbor současného stavu sběru dat z měřicího polygonu vrtů na VŠB-TU Ostrava

V rámci zkoumání vnitřních a vnějších zdrojů horninového masivu byly vybudovány dva výzkumné měřicí polygony, tj. malý výzkumný polygon (MVP) a velký výzkumný polygon (VVP). MVP zkoumá změny chování hornin v okolí vrtů, kde jsou změny teplot způsobeny odběrem tepla ze dvou vrtů připojených na tepelná čerpadla. Měření VVP je zejména určeno pro sledování vlivu velkého odběru energií z horninového masivu a podílí se na vytápění a klimatizování Nové auly + CIT VŠB-TUO.

Obr. 2.1 Rozložení obou polygonů a jednotlivých vrtů



Zdroj: Vyhodnocení projektu – Výzkum teplotních změn horninového masivu (ochlazování – oteplování) při aplikaci tepelných čerpadel v lokalitě Nové auly + CIT VŠB-TU Ostrava (prof. Ing. Tomáš Čermák, CSc., 2007-2009)

2.1 Malý výzkumný polygon MVP

Sběr dat je prováděn na devíti vrtech, z toho jsou čtyři technologické a pět monitorovacích. Mimo tyto vrty jsou dále monitorovány tlaky, průtoky čerpadel aj. Obr. 2.3 znázorňuje rozložení a strukturu vrtů.

Řídicí systém, který zajišťuje sběr dat, je PLC X20CP1484 firmy B&R Obr. 2.2. Obsahuje procesor Celeron 266 MHz a je vybaven následujícími periferiemi: 1x RS232, 1x ETHERNET, 1x POWERLINK V1/V2, 2x USB, 1x X2X Link.

Obr. 2.2 PLC X20CP1484



Zdroj: B&R

Pomocí přídavných rozšiřujících modulů zpracovává data teplotních čidel PT 1000 umístěných v jednotlivých vrtech. Jedná se hlavně o čidla, která disponují výstupem v podobě proměnné velikosti odporu. Údaje z těchto čidel zpracovávají moduly X20AT4222. Zásuvné moduly jsou přímo určené pro PT 100 a PT 1000 a nastavitelné pro dvou, tří, čtyř-vodičové připojení s nastavením kanálu. D/A převodník je 16 bitový a výsledek je uložen jako INT, popřípadě (UINT). Rozsah měřené resistance se pohybuje od 0.1 do 4500Ω /0.05 až 2250Ω. Druhá část použitých odporových teplotních senzorů PT 1000 disponuje digitálním výstupem na sběrnici RS232. Digitální výstupy z těchto čidel zpracovává přímo procesorová jednotka X20CP1484.

V PLC je nastavena komunikace pro OPC server, který přeposílá získaná data dále ke zpracování. Vzdálený přístup k těmto datům je nastaven v PVI OPC Configuratoru. V tomto programu je nastavena IP adresa automatu a nadefinované proměnné. OPC (OLE for Process Control) je standard průmyslové komunikace, který určuje společné rozhraní pro komunikaci mezi dvěma či více zařízeními určenými pro monitorování a řízení technologických

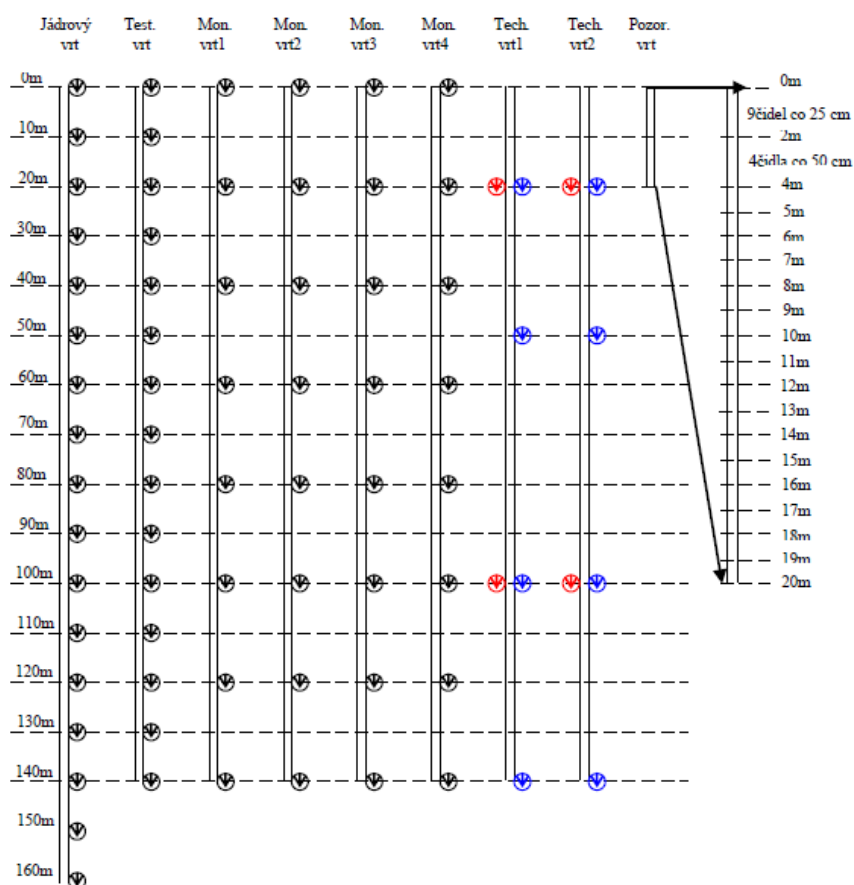
procesů. Má za úkol zabránit závislosti daného monitorovacího nebo řídicího softwaru na výrobci hardwaru. Tato norma umožňuje programátorům větší možnosti v rámci OPC klientů.

Dále se data ukládají na FLASH paměť uvnitř PLC. K těmto datům je přístup pomocí FTP protokolu a jsou uložena v CSV formátu (CSV formát je složen z řádků, ve kterých jsou jednotlivé položky odděleny znakem čárka).

Součástí řídicího systému je i osobní počítač komunikující s hlavním PLC pomocí komunikační sběrnice Ethernet. Na tomto PC je nainstalováno vývojové prostředí B&R Automation Studio, které slouží pro vývoj a ladění samotné řídicí aplikace.

V rámci vývoje celého systému je v učebně na NK317 vyhrazen počítač, který bude sloužit pro úložiště, správu dat a webovou vizualizaci.

Obr. 2.3 Schéma teplotních čidel v MVP



Zdroj: Vyhodnocení projektu – Výzkum teplotních změn horninového masivu (ochlazování – oteplování) při aplikaci tepelných čerpadel v lokalitě Nové auly + CIT VŠB-TU Ostrava (prof. Ing. Tomáš Čermák, CSc., 2007-2009)

2.2 Velký výzkumný polygon VVP

VVP obsahuje 10 provozních vrtů a 5 speciálních (monitorovacích) vrtů. Provozní vrty jsou realizovány ve dvou rovnoběžných řadách s označením V071, V073, V075, V077 a V079 v jedné řadě (jižní profil) a V081, V082, V084, V086, V088 v souběžné řadě (severní profil). Znázornění na Obr. 2.1. Na vstupní straně byla osazena čidla v hloubce 20, 50, 100 a 140 m, na výstupní straně v hloubce 20 a 100 m. Speciální (měřicí) vrty jsou označeny MV01 až MV05 a na stejném obrázku mají červené označení. Zde se nachází čidla v úrovních 20, 50, 100 a 140 m. Poslední nezmíněný vrt je označen KV01 (kontrolní hydrogeologický vrt).

Obr. 2.4 Rozvodnice v šachtici č. 4. VVP



Zdroj: Vyhodnocení projektu – Výzkum teplotních změn horninového masivu (ochlazování – oteplování) při aplikaci tepelných čerpadel v lokalitě Nové auly + CIT VŠB-TU Ostrava (prof. Ing. Tomáš Čermák, CSc., 2007-2009)

Hlavní měřicí jednotkou je ICPDAS s označením I-8KE8, ve kterém jsou zasunuty moduly s označením I-87013 (parametry modulů v Tab. 2.1). Pro konfiguraci těchto I/O modulů je poskytován program zvaný VxComm (Virtual Communications Ports) založený na DCOM protokolu, který zpřístupňuje I/O moduly po ethernetu. Pro tyto tři jednotky jsou vytvořeny virtuální COM porty.

Teplotní čidla jsou taktéž zapojena čtyř-vodičovým způsobem na vstupní svorky měřicích modulů. Celkově se měří 80 teplot a jedna měřicí karta dokáže obsloužit čtyři teplotní

čidla. Je tedy použito 20 teplotních modulů, které jsou zasunuty do ICPDAS jednotek a komunikují po ethernetu. Každá jednotka obsahuje osm měřicích modulů. 80 čidel je zapojeno do tří kusů jednotek, které jsou dále propojeny do sítě přes ethernet. Jednotky tedy zpracovávají analogové hodnoty a dále je přeposílají po virtuálních portech do počítače, kde je v provozu OPC server.

OPC server zpřístupňuje data ke zpracování OPC klientům. Jako klient může být použit jakýkoliv OPC klientský software jiných výrobců. Protože OPC je standard, ze kterého musí všechny programy vycházet stejně. Monitorovací PC, které se nachází v místnosti strojovny tepelných čerpadel v Nové aule, je zapojeno do switchu, kde jsou připojeny všechny tři jednotky. Na tomto PC pracuje SCADA systém od firmy Promotic, který zpracovává teplotní údaje. Tento program také obsahuje OPC klienta, kde jednoduché nastavení umožňuje zpracovávání dat.[1].

Tab. 2.1 Specifikace zásuvných modulů I-87013 [2]

Analogový vstup	počet kanálů	4 RTD vstupní kanály
	typy vstupu	2-vodičové,
		3-vodičové,
		4-vodičové RTD
	čidla	Pt100, Pt1000, Ni120
	vzorkování	10 vzorků/s max.
	-3dB BandWidth	15.7Hz
	přesnost	+/-0.1%
	Common Mode Rejection	150dB @50/60Hz
	Normal Mode Rejection	100dB @50/60Hz
Izolační napětí	3000Vrms	
LED Displej	1 LED indikace zapnutí/komínikační indikátor	
	8 LED High/Low alarmových signálů	
Spotřeba	Maximum: 0.7W	

Zdroj: B&R

2.3 Účel a rozdíly sběru dat u obou polygonů

Kapitoly 2.1 a 2.2 rozebírají a popisují skutečné použité technické prostředky a jejich rozložení v prostoru. Tato kapitola pohlíží na polygony obecněji.

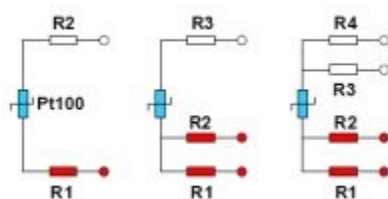
Význam sběru dat spočívá v posouzení vlivu vnitřních a vnějších zdrojů na tepelnou bilanci v horninovém masivu z hlediska rychlosti obnovitelnosti tohoto alternativního zdroje energie. MVP a VVP jsou dva různé systémy, jejichž sběr dat je zajištěn různými systémy určenými pro měření teplot, viz 2.1 a 2.2. Získaná data, ukládaná v čase, poté slouží k vyhodnocení tepelné bilance v hornině. Tyto výsledky měření mohou dále sloužit pro podobné projekty s podobným složením horniny, jako mají MVP a VVP.

Oba polygony jsou vytvořeny nezávisle na sobě, přičemž plní stejnou funkci-zjišťování tepelných hodnot a jejich uložení v čase. Jejich úložiště dat pracují na různých softwarových prostředcích. Nahlížení na tyto data lze provádět pouze odděleně. Data v MVP jsou uložena na FTP serveru ve formátu CSV. VVP má svou vlastní plnohodnotnou databázi.

2.4 Popis čidla PT 1000

Existuje dvou-vodičové, tří a čtyř-vodičové zapojení. V obou měřicích systémech jsou zakomponovány zapojení se čtyřmi vodiči z důvodu nejvyšší přesnosti. Obr. 2.4 znázorňuje princip propojení PT čidel těmito způsoby.

Obr. 2.4 Dvou, tří, čtyř – vodičové zapojení



Zdroj: vlastní

Obecně by se teplotní čidlo PT 1000 dalo charakterizovat následovně. Na keramický podklad se napařuje tenká vrstva platiny, která je pomocí laseru přesně nastavená na jmenovitou hodnotu PT 1000, popřípadě jiného typu čidla jako jsou PT 100 nebo PT 500. Dále je nanášena izolační vrstva, která chrání platinu před nepříznivými vlivy. PT 1000 má rozměry 2,0 x 5,0 x

1,0 mm a čidla jsou standardně vyráběna ve dvou třídách přesnosti A a B. Dále je přiložena tabulka Tab. 2.2 s parametry PT 1000. Informace čerpané z [7].

Tab. 2.2 Parametry PT 1000

Snímací prvek	Tenkovrstvý platinový odpor
Maximální rozsah pracovních teplot	-50 až 400 °C *
Odpor při 0 °C	1000 Ω
Dlouhodobá stabilita odporu	0,03% po 1000 hod při teplotě 400 °C
Doporučený / maximální ss měřicí proud	0,3mA / 1mA

Zdroj: http://www.sensit.cz/ke-stazeni/soubory/Pt1000_3850.pdf

*Rozsah dle konstrukce a technologie

$$R = 1000(1 + At + Bt^2 + C(t - 100)t^3) \quad \text{rozsah teploty -50 až 0 °C} \quad (1)$$

$$R = 1000(1 + At + Bt^2) \quad \text{rozsah teploty 0 až 400 °C} \quad (2)$$

kde:

$$A = 3,9083 \cdot 10^{-3} \text{ } ^\circ\text{C}^{-1} \quad (3)$$

$$B = -5,775 \cdot 10^{-7} \text{ } ^\circ\text{C}^{-2} \quad (4)$$

$$A = -4,183 \cdot 10^{-12} \text{ } ^\circ\text{C}^{-4} \quad (5)$$

Třída přesnosti:

Tab. 2.3 Třída přesnosti

Čidla jsou vyráběna ve dvou základních třídách přesnosti, vyjádřenými těmito vztahy:

Třída A	$\Delta T = \pm (0,15 + 0,002 \cdot t) \text{ ve } ^\circ\text{C}$
Třída B	$\Delta T = \pm (0,30 + 0,005 \cdot t) \text{ ve } ^\circ\text{C}$

* |t| je absolutní hodnota teploty

Teplota [°C]	Odpor [Ω]	Třída A		Třída B	
		ΔT [°C]	ΔR [Ω]	ΔT [°C]	ΔR [Ω]
-50	803,06	± 0,25	± 0,99	± 0,55	± 2,18
0	1000,0	± 0,15	± 0,59	± 0,30	± 1,17
100	1385,06	± 0,35	± 1,33	± 0,80	± 3,03
200	1758,56	± 0,55	± 2,02	± 1,30	± 4,78
400	2470,92	± 0,95	± 3,27	± 2,30	± 7,94

Zdroj: http://www.sensit.cz/ke-stazeni/soubory/Pt1000_3850.pdf

3. Analýza možností přenosu měřených dat z malého a velkého měřicího polygonu na databázový server. Analýza možností internetové vizualizace těchto dat.

PLC malého a ICPDAS velkého výzkumného polygonu se nacházejí každý v jiné budově. Pro potřebu sjednocení dat na jedno místo, do jedné databáze, je třeba vzít v úvahu možnosti obou programovatelných jednotek. Důležitou otázkou v přenášení dat je konzistentnost těchto dat, správně navržená databáze nejen vnitřní struktury, ale i vnější komunikace databáze s okolím. U VVP je nutná konfigurace systému do sítě.

Podkapitole 3.2 analyzuje možnosti internetové vizualizace. Zabývá se skriptovacím jazykem PHP a technologií ASP.NET.

3.1 Možnosti přenosu měřených dat do databáze

Stav v dnešní průmyslové praxi s velkým přehledem nahrává OPC použití. Většina výrobců PLC má integrovanou podporu OPC serveru. Na obou polygonech jsou zařízení, která OPC standard podporují. Níže je jako první možnost popsán princip, problematika a specifikace OPC.

OPC specifikace je založena na technologii OLE, COM a DCOM, kterou vyvinula společnost Microsoft. Dnes se na toto téma zaměřila dobrovolná organizace OPC Foundation se sídlem ve Scottsdale, ve státě Arizona v USA. Organizace sdružuje přes 500 členů z významných firem na celém světě, které se zabývají monitorovacími, vizualizačními a podobnými aplikacemi. Jejich domovská stránka je na adrese www.opcfoundation.org. Mezi známé firmy, které se sdružují do této organizace, patří Siemens, B&R, Rockwell Software, Honeywell aj.

Hlavní důvod spočívá v definování standardů pro přenos dat mezi průmyslovým a počítačovým prostředím. Nejvíce používaná a nejznámější specifikace se nazývá OPC Data Access, používaná k přenosu Real-Time dat. OPC Data Access je zdlouhavý název, proto se v průmyslové praxi uchytlo jednodušší označení, a to OPC DA [3].

Ostatní specifikace [3]:

- OPC Alarms & Events (OPC AE)
- OPC Batch
- OPC Complex Data
- OPC Data Access (OPC DA - nejčastěji používaná specifikace)
- OPC Data Exchange (OPC DX)
- OPC Historical Data Access (OPC HDA)
- OPC Security
- OPC XML-DA

OPC server

OPC server je software, který komunikuje s protokolem PLC (např. MPI) a přeposílá data ve formátu OPC nadřazeným aplikacím. Výrobci programovatelných automatů si vytvářejí vlastní OPC servery, jejichž spuštění a nastavení je velmi jednoduché. V zásadě jsou podpořeny intuitivními návody pro zprovoznění. Existují také OPC servery, které jsou schopny se připojit přímo k mnoha programovatelným automatům různých výrobců. Příkladem je program KepServer, který je schopen obsloužit desítky různých typů PLC. Mezi nejznámější patří:

- Siemens MPI / PPI / TCP/IP
- Modbus Serial / Ethernet
- Allen Bradley DF1 / Ethernet / DH+
- Mitsubishi FX / FX Net / Ethernet / Serial
- BACnet zařízení
- Yokogawa Darwin Ethernet / Serial / DX / DX
- a další...

OPC klient

OPC klient je software, který přijímá data z OPC serveru. Data jsou dále poskytována monitorovacím a vizualizačním aplikacím, známým pod názvem HMI/SCADA systémy. Složitost programu klienta a serveru je zásadně rozdílná. Server slouží pouze pro základní přeposílání dat. Program klienta musí být variabilní a přizpůsobivý využitelnosti dat. Například OPC klient programu OPC DataHub je schopen webové vizualizace, ukládání dat do databáze, posílání e-mailu a mnoho dalších možností. Tato aplikace je však značně specifická těmito

vlastnostmi, protože mnoho klientů se zabývá jiným zpracováním, např. zpracováním dat, vizualizací, zobrazováním grafů atd.

OPC jako celek je velmi silný nástroj pro přenos dat. Přináší zjednodušení konfigurace a úspory času. Programátor se nemusí zabývat, jakého klienta musí použít, ale použije takového, se kterým má dobré zkušenosti.

Světově nejznámější SCADA systémy s podporou OPC klienta [3]:

- InTouch (Wonderware)
- WinCC (Siemens)
- iFix (Intellution)
- Genesis 32 (Iconics)
- CitectScada (Citect)
- RSView32 (Rockwell Software)
- Lookout (National Instruments)

Čeští výrobci SCADA systémů s podporou OPC klient [3]:

- Aspic (Merz s.r.o.)
- ControlWeb (Moravské přístroje a.s.)
- Promotic (MICROSYS, spol. s r.o.)
- Reliance (GEOVAP, spol. s r.o.)

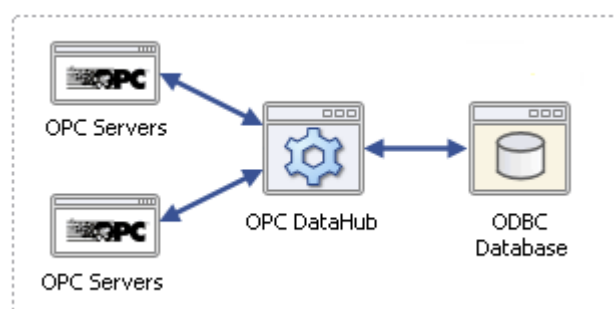
Tento způsob přenosu dat ještě nestačí k ukládání dat do databáze. Je třeba zajistit propojení OPC a databáze. Zde se využívá ODBC (Open Database Connectivity) . Jedná se o standardizovaný způsob komunikace s databází. Snaha ODBC spočívá v nezávislosti přístupu různých programovacích jazyků.

Klientský software přistupuje k OPC serveru a převádí data na tento standard. Taktéž se jedná o silný nástroj, kde se k databázi může připojit spousta různých klientů. Pro analyzování těchto možností je popsáno několik softwarů (podkapitola 3.1.1), které tyto možnosti ukládání dat do databáze umožňují.

3.1.1 OPC standard a přístup do databáze

Jako první program, který ukládání do databáze umožňuje je DataHub. Tento program vyvíjí americká firma Cogent Real-Time Systems. Má implementováno mnoho způsobů použití. V hlavní řadě obsahuje OPC server a klienta. Tato problematika byla popsána výše. Navíc obsahuje funkci OPC Data Logging, kvůli které je software vhodný pro ukládání dat z OPC serveru do databáze prostřednictvím ODBC.

Obr. 3.1 OPC DataHub propojení serveru s databází



Zdroj: http://www.opcdatahub.com/Features/OPC_Logging.html, vlastní úprava

Obr. 3.1 zobrazuje princip přístupu k různorodým systémům. DataHub je softwarový prostředek, který obsluhuje standard na vstupu a standard na výstupu a slouží jako prostředník mezi nehomogenními systémy.

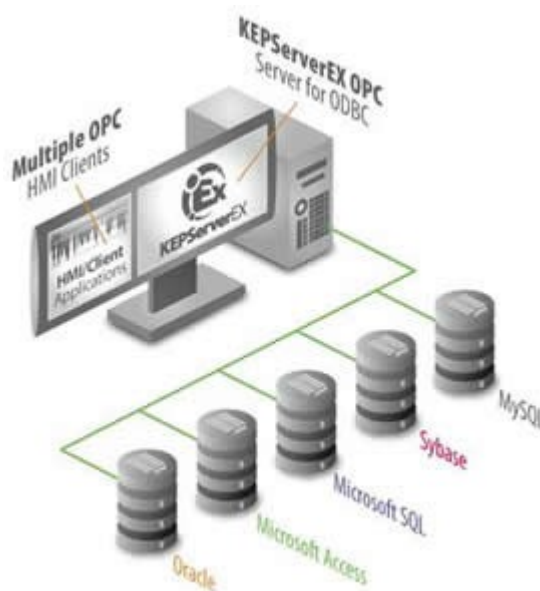
Mimo základní potřebnou funkci, kdy DataHub přenáší data z OPC do databáze, je schopen provádět i další užitečné funkce.

- Přenášet data mezi OPC servery – známé pod názvem OPC Tunnelling
- Je schopen provádět různé procesy (násobení, dělení, průměrování atd.) a následně data poslat dále
- Má funkci webového serveru, dokáže zobrazovat OPC data na internetu – Web server
- A v neposlední řadě umí číst z OPC a zapisovat do databáze přes ODBC – Data Logging

Další možností, která se nabízí z obrázku Obr. 3.2 je použití softwaru od firmy Kepware Technologies sídlící v Portlandu v USA. Princip přenosu dat do databáze je podobný. Nastává však otázka, zda by tento model mohl být použit pro potřeby přenosu dat. Kepware OPC server je aplikace sama o sobě. Pro potřeby ukládání do databáze je třeba dokoupit aplikaci podpůrnou, která se nazývá ODBC Client driver. Toto spojení potom funguje jako jeden celek. Záleží jen na OPC serveru, na který PLC systém je schopen se připojit. Podpora ICPDASu a B&R

programovatelných automatů není podporovaná u tohoto softwaru. Přitom seznam podporových systému je velice obsáhlý. Podle zjištěných informací se Kepware Technologies do budoucna nevyjádřil o poskytování driveru pro podporu B&R automatů a podpora méně známého ICPDASu je taktéž do budoucna vyloučena.

Obr. 3.2 KepServer a ODBC Client driver

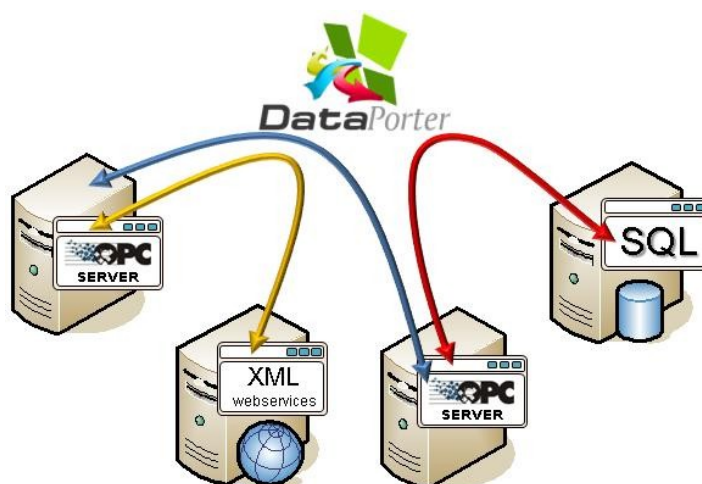


Zdroj: http://www.kepware.com/Spec_Sheets/ODBC_Client_Driver.asp

Další použitelný software se nazývá DataPorter od firmy CommServer sídlící v Lodži v Polsku. Jejich sortiment softwaru je velmi rozsáhlý, např. OPC server, CommServer UA pro velké distribuované systémy, OPC Viewer atd..

DataPorter je nezávislý balíček, navržen pro průmyslové aplikace. Jeho schopnosti spočívají v převodu dat mezi populárními standardy, jako je OPC, XML a SQL. V našem případě se nastavuje přístup do OPC Serveru a do databáze. DataPorter potom pracuje jako prostředník mezi těmito odlišnými systémy. Software je principiálně velice podobný DataHubu.

Obr. 3.3 DataPort a znázornění převodu dat různých formátů



Zdroj: <http://www.commsvr.com/Products/OPC/DataPorterDescription.aspx>

DataPorter - má také mnoho zajímavých funkcí, např.

- Přenáší data mezi OPC servery
- Je schopen provádět různé procesy (násobení, dělení, průměrování atd.) a následně data poslat dále
- Má funkci webového serveru, dokáže zobrazovat OPC dat na internetu – Web server
- Číst, zapisovat do SQL databáze

Software je dostupný ve dvou verzích:

- DataPorter Standard – zpracování dat mezi OPC servery
- DataPorter Professional – všechny výše popsané možnosti

Jako poslední je představen OPC klient od firmy OPC Systems. Mezi jejich nepřeberným množstvím softwaru je jeden, který se nazývá OPC DATABASE.NET. Poskytuje Data logging do mnoha databází (Access, Oracle, mySQL aj.) Další dobrou vlastností při ztrátě spojení je využití bufferu, který dočasně loguje data do mezipaměti a po obnovení spojení se přenos dat obnoví. Jeho cena se odvíjí od počtu použitých proměnných.

Je tedy na výběr „prostředník“, jež bude plnit výše popsanou funkci, kterého je možno vybrat z těchto zjištěných informací. Nejzajímavější software nabízí CommServer a firma Cogent RT Systems. Další nevyhovuje z důvodu nepodporování PLC automatů na obou polygonech a druhý je cenově nevýhodný zaplacením licence podle počtu použitých proměnných. Použitým softwarem se stal OPC DataHub. Není omezen, co se týče počtu

použitých proměnných. Nahrává to dalšímu použití např. zapojením do systému dalšího měřicího polygonu. Další výhodou tohoto programu je dobře zpracovaný návod.

3.1.2 Jiné standardy s možností přístupu do databáze

Oba měřicí polygony disponují OPC technologií. Jiné další společné periferní vlastnosti nemají, mimo přídavných modulárních komponent, které by stávající schopnosti měřicího systému rozšiřovaly.

B&R patří mezi známější firmy v oboru měření a řízení. Není divu, že přichází s mnoha způsoby komunikace. Např. Ethernet TCP/IP, který přináší protokoly HTTP, FTP, SMTP. Jejich použití je tak vhodné i pro správu dat v kancelářských aplikacích. Je tedy možnost analyzovat data přímo v internetovém prohlížeči, na FTP serveru anebo přes e-maily [6]. Ne všechny možnosti jsou však vhodné pro propojení s databází. Například e-maily a webové stránky jsou vhodné jen pro informativní účely.

FTP server a jeho dosavadní nastavení, by bylo možné použít. Mezi FTP serverem a databází je potřeba prostředníka, který by kopíroval, doplňoval jednotlivá data a přeposílal do databáze. Je nutné, aby tento programový prostředek byl postaven na míru, protože struktura dat na FTP serveru procházela také svým vývojem a každá změna (zásah) struktury by měla za následek i změnu (úpravu) prostředníka. Jednotlivé záznamy jsou vytvářeny jako měsíční soubory. Z tohoto faktu vyplývá, že vývoj tohoto prostředku by trval značně dlouho. Taktéž uvážení ukládání v čase je na místě. „Prostředník“ jehož cílem by bylo přeposílání dat, by musel mít nastavený čas provedení akce. Důležitější však je nastavení ukládání dat na FTP server (záležitost naprogramování PLC automatu), kde tento čas nejde prostředníkem ovlivnit. Další poznatek je, že čas ukládání na FTP server musí být menší jak ukládání do databáze přes prostředníka. Tento způsob řešení problému je složitější.

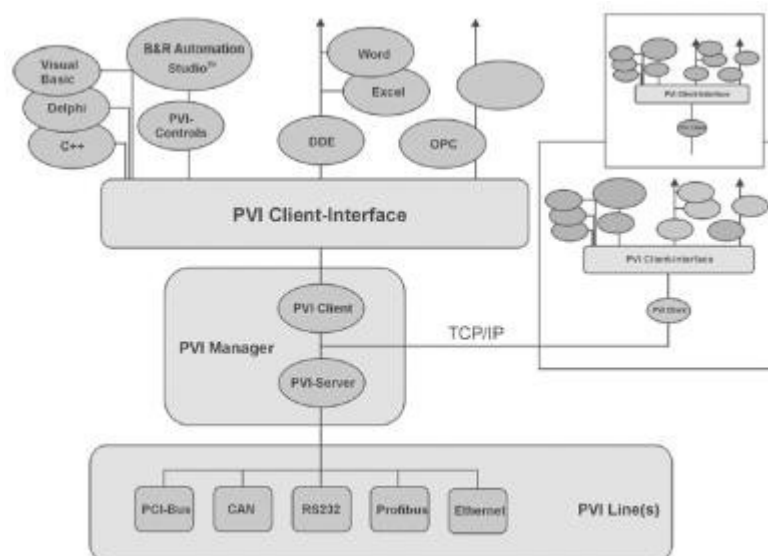
B&R dále poskytuje PVI (Process Visualization Interface). Pomocí tohoto softwaru jsou data dosažitelná v nástrojích MS Windows. Výstup je prezentován přes DDE server, OPC server, web server, DLL knihovnu. Jsou tedy přístupná i přes MS Office. Nástroje tak mají schopnost jednoduchého propojení s databází [6].

Obr. 3.4 názorně ukazuje možnosti propojení. Na výběr je i z jiných možností, např. použití C++, Delfi nebo Visual Basic. Implementace je zde podobná jako při použití FTP.

OPC standard ze všech dalších možností vychází nejlépe, je totiž navržen pro práci s daty v reálném čase. Dosažitelnost dat je prakticky okamžitá. Jakoby měnící se data čekala na „otevřené dveře“ (při zavřených jsou nepoužita). Data jsou tedy čitelná vždy a záleží jen na nastavení komerčního produktu (Dataporter, DataHub, Kepware OPC server-jeho ODBC Client

driveru a dalších), kdy je libo data přeposlat do databáze. Možnost spolehnout se na tyto produkty je vysoká. Programátor se nemusí zabývat vlastním vývojem potřebného mezičlánku.

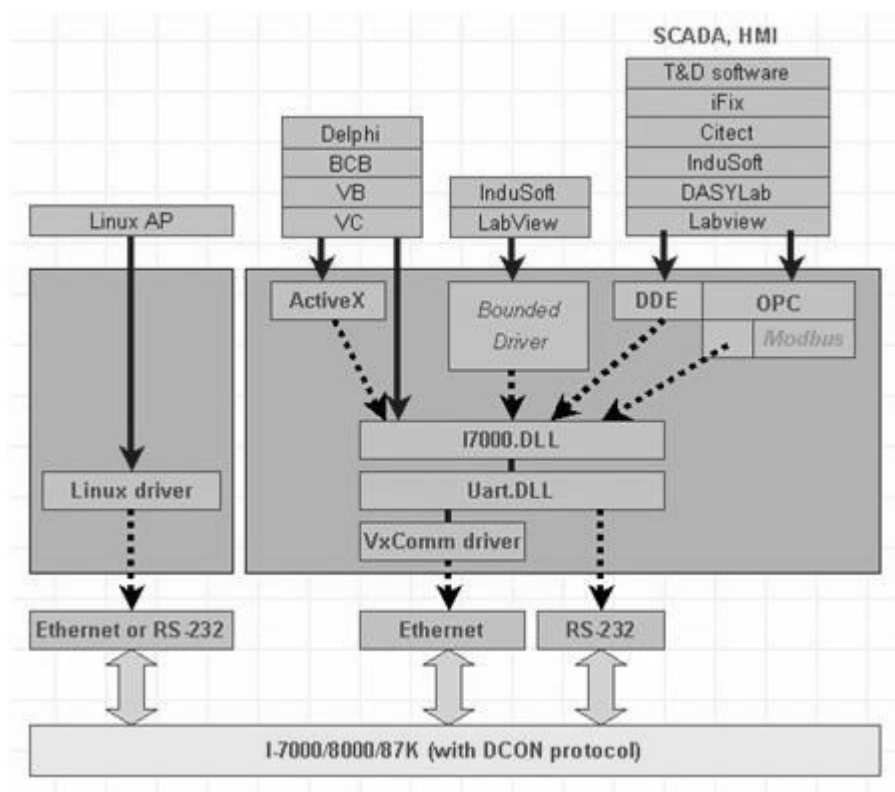
Obr. 3.4 Schéma B&R Automation Net



Zdroj: http://www.odbornecasopisy.cz/index.php?id_document=28783

ICPDAS je také schopen pracovat s mnoha programy, jak je znázorněno na Obr. 3.5. Propojení je poskytováno přes protokol DCOM. Zde jsou vytvořeny drivery pro DLL knihovny, ActiveX prvky, Labview, Indusoft a Linux driver. Použitelnost je rozsáhlá. Výrobce v manuálu názorně ukazuje použití těchto driveru. Pro účely polygonů, kdy je potřeba data ukládat do databáze, vyhovuje pouze OPC a DDE, pokud budeme uvažovat snadný návrh propojení s databází. Ve Windows je s přístupem k protokolu DDE například MS Office. Tyto nástroje, stejně jako u B&R, jsou schopny se do databáze připojit. I zde platí, že nejjednodušším použitím je OPC protokol.

Obr. 3.5 ICPDAS a vnější komunikace



Zdroj: 8430_8830_8ke4_8ke8_manual [CD-ROM]

3.1.3 Databáze

Databáze je uspořádaná množina informací uložená z pravidla na fyzické paměti. U velkých databázových systémů se jedná hlavně o velkokapacitní pevné disky (HDD). Data jsou strukturována v tabulkách a mají možnost reference mezi tabulkami. Tahle relace mezi tabulkami zavedla pojem relační databáze. Pod tímto pojmem se dnes vyskytuje většina databázových systémů, jako jsou MySQL, MSSQL(Microsoft SQL Server), Oracle a další [5].

V první řadě je tedy nutné vybrat takovou relační databázi, která bude vhodná pro potřeby sběru dat VVP a MVP. Úlohou potřebné relační databáze je spolehlivé ukládání dat, správa dat (vykonávání příkazů) ve smyslu jejich analyzování, třídění a vyhledávání. Funkce mazání a editace dnešní plnohodnotné databázové systémy taktéž umožňují, avšak pro potřeby měřicích polygonů nejsou potřeba (nežádoucí).

Neméně důležité informace se týkají tabulek a jejich struktury. Každý řádek v tabulce je pojmenován jako datový záznam charakterizovaný datovým typem. Zpravidla se definují

datové typy pro každý sloupec v tabulce [4]. Níže je znázorněn návrh definice tabulky z MVP vrtu TC1, na němž je ukázána struktura tabulky a příslušné datové typy.

Tab. 3.1 Návrh datových typů

jméno sloupce	datový typ	povolit NULL
čas	Datetime	✓
TC1_20T_m	Int	✓
TC1_100T_m	Int	✓
TC1_20S_m	Int	✓
TC1_100S_m	Int	✓
TC1_140S_m	Int	✓

Zdroj: vlastní

Jméno sloupce „čas“ je označeno datovým typem „Datetime“. Přidávané záznamy tohoto typu přesně definují celý čas od roků po vteřiny. Pro další položky je možnost výběru INT, anebo REAL. Oba datové typy mají stejnou velikost s tím rozdílem, že REAL ukládá hodnotu v desetinném vyjádření. Hodnoty v této položce budou reprezentovat měřené teploty, tlaky aj. V úvahu je brána skutečnost, že hodnoty z OPC serveru z MVP přichází ve formě celočíselné hodnoty. Poslední sloupec „povolení NULL“ znamená, že v záznamech mimo celočíselná čísla se může zobrazit i hodnota NULL (neznámá hodnota). Tato možnost poslouží při poruše-jednodušší analýza problému.

- Datetime 8bajtů (YYYY-MM-DD HH:MM:SS)
- Int 4bajty (-2 147 483 648 až 2 147 483 647)
- Real 4bajty (1E-37 až 1E+37)

Velikost jednotlivých tabulek značně ovlivňuje rychlost hledání potřebných informací. V databázových systémech se hojně využívá indexu neboli klíče. Index jednoznačně identifikuje každý záznam v tabulce [5]. V návrhu index není zahrnut, protože jeho použití za prvé značně navyšuje velikost datového souboru a za druhé je jednoznačnost každého záznamu přesně dána použitým typem „datetime“. Žádný záznam nemá stejný čas.

Všeobecně tabulky ve většině případů obsahují data v náhodném pořadí. Návrh ukládání záznamu je navržen po pěti minutách, z čehož vyplývá časová posloupnost záznamů. Záznamy jsou časově řazeny za sebou, ale pro efektivní zpracování, jako je výběr určitého časového

úseku (týdne, dne, hodin), je třeba vytvořit dotaz. Ve výsledku se vytvoří opět tabulka uložena v operační paměti. Pro vytvoření dotazu se používá instrukce SQL příkazů. SQL (Structured Query Language - strukturovaný dotazovací jazyk) je standardizovaný jazyk pro formulaci dotazů v databázových systémech. Existují modifikace jako například v MSSQL, který používá T-SQL (Transact-SQL) [4]. Všeobecně se ve všech systémech pracuje s příkazy stejně. Níže je zobrazen příklad, který začíná příkazem SELECT. Pomocí této funkce získáme obsah tabulky. TOP 5 vybere pět záznamů z (FROM) tabulky TC1, řazených (ORDER BY) podle času od nejnovějšího záznamu směrem dolů.

"SELECT top 5 * FROM TC1 ORDER BY čas DESC" (6)

Tab. 3.2 Návrh tabulky TC1

čas	TC1_20T_m	TC1_100T_m	TC1_20S_m	TC1_40S_m	TC1_100S_m
25.3.2010 11:50	54	17	56	47	47
25.3.2010 11:45	54	17	56	47	47
25.3.2010 11:40	54	16	56	47	47
25.3.2010 11:35	53	17	55	47	47
25.3.2010 11:30	53	17	55	46	46

Zdroj: vlastní

Uvedený postup popisuje vyčítání hodnot z databáze pomocí SQL příkazů. Hlavní využití je v klientských softwarech, které k těmto datům budou přistupovat. Zapisování dat bude provádět další klientský software, jehož vlastností je převod OPC dat do databáze přes ODBC standard, anebo SQL příkazy.

Mimo ODBC jsou použitelné další rozhraní poskytující přístup k datům, např. OLE DB (Object-Linking and Embedding Database) nebo ADO.NET, který poskytuje přímou komunikaci se serverem bez dalších programovatelných vrstev. Podpora je jen u málo databází. Výrobci klientských systémů dávají přednost nejpoužívanějšímu standardu, kterým je ODBC. Byly rozebrány vlastnosti databáze, jejich potřebná struktura a přístupy do databáze přes určité standardy.

Výrobci se drží zavedených postupů. Je tedy otázkou, který databázový systém zvolit. Principy práce s databázemi jsou velmi podobné, různá jsou ale prostředí pro správu. Mezi nejznámější patří Oracle, MySQL a MSSQL.

MSSQL má dlouhou historii. Jeho první databázový systém pochází z roku 1989. Mezi nejnovější patří MSSQL Server 2005 a MSSQL Server 2008. U databázových systémů od firmy

Sun Microsystems je nejnovější a hojně využívaný systéme hlavně v kombinaci s webovým serverem MySQL 5.5. Oracle nabízí aktuální verzi 11g a má velké zastoupení na trhu.

Jako databázový software byl vybrán produkt MSSQL Server 2008 Express. Jeho nástroj pro správu databáze se nazývá SQL Server Management Studio Express. Express instalace je zdarma a velikost databáze 4GB. Podle naměřených hodnot vyplývá, že tato databáze je svou velikostí vyhovující. Roční přírůstek jednoho polygonu je cca 60MB. Měření bylo provedeno na MVP, kde je hodnot více.

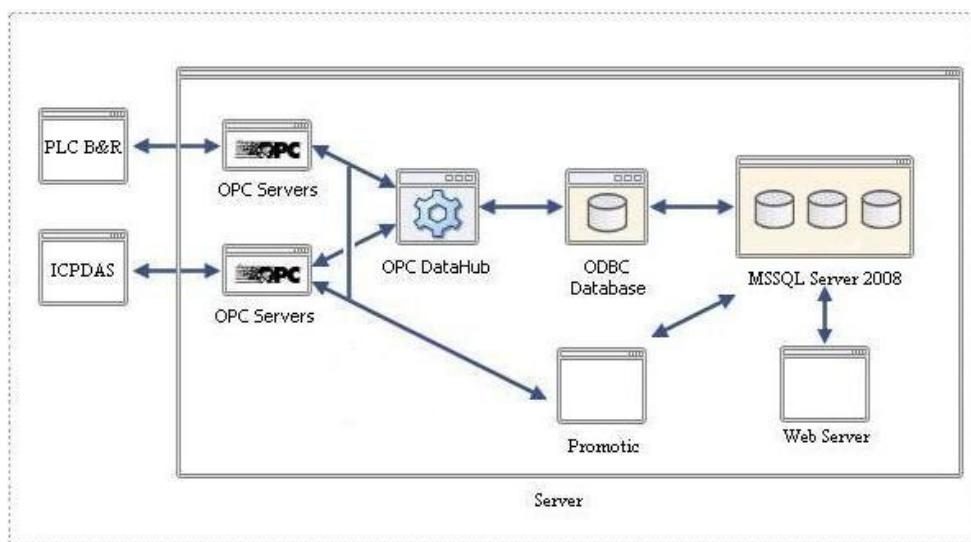
3.1.4 Databáze a zvolený způsob ukládání dat

Zakoupená licence DataHubu nastínila směr vývoje celého systému. Společně s databází od Microsoftu bude server plnit úlohu serveru webového a databázového.

Neméně důležité informace o OPC serveru, DataHubu a MSSQL, bez kterých by vizualizace nemohly fungovat, popisují všechna jednotlivá fyzická propojení.

Jednotlivé programy poskytují ostatním programům data přes určitá rozhraní. Pro ilustraci slouží Obr. 3.6. Vzhledem k tomu, že jednotlivé nastavení všech programů jsou velice obsáhlá, jsou vytvořeny návody pro každý program, které se nachází v přílohách. Návody se zabývají orientací v jednotlivých programech společně s každým nastavením.

Obr. 3.6 Fyzická komunikace jednotlivých programů



Zdroj: vlastní

3.2 Možnosti internetové vizualizace

Přenosem dat přes OPC rozhraní do databáze a návrhem vnitřní struktury databáze je potřeba archivovaná data určitým způsobem prezentovat. Přímý náhled, analyzování dat v databázi přes podpůrné programy (SQL Server Management Studio), je vhodný pouze pro znalé. Uživatel se nemusí učit znalostem SQL příkazů. Je odkázán na jednoduchou práci s předdefinovanými funkcemi, které tyto příkazy pod sebou ukrývají. Proto je nejefektivnějším prostředkem zobrazování upravených dat na webu, kde je práce s daty převedena do uživatelsky příjemné podoby. Další výhodou je zobrazování dat v síti nebo na internetu, tj. jednoduchý přístup odkudkoliv.

Je nutné zajištění přístupu do databáze a následné vyčtení dat, které požaduje uživatel. Mimo vyčítání je také nutná úprava vyčtených dat do konečné podoby a prezentace ve formě tabulky a grafu znázorňující vybrané údaje.

Zde se naskytují dvě použitelné možnosti. Jsou to PHP a ASP.NET. Oblíbenost obou technologií tkví především v jednoduchosti použití. Jejich návrh využívá více programovacích jazyků. Pro vývojáře to znamená variabilnější syntaxe kódu. Obě technologie jsou především určeny pro dynamické internetové stránky. Další společná vlastnost je vykonávání dynamických akcí na straně serveru, kdy je ke klientovi odeslán požadovaný konečný výsledek.

PHP je skriptovací jazyk podporující internetové protokoly (HTTP, SMTP, FTP a jiné). Internetové protokoly jsou standardem, ale pro práci s databázemi jsou důležité přístupy k databázovým systémům. I tady PHP nabízí potřebné knihovny (MSSQL, MySQL, ODBC, Oracle). Nejrozšířenější využití jazyku se nachází ve webových aplikacích, kdy je ve spojení s webovým serverem Apache a databází MySQL.

ASP.NET je software od firmy Microsoft. Vychází ze starší technologie zvané ASP. Mají však málo společného. ASP.NET je založen na CLR (Common Language Runtime). Z toho vyplývá výhoda v programování více jazyky, např. C++, C#, Visual Basic.NET. Jako konečná aplikace pracuje mnohem rychleji než skriptovací jazyky. Kódy jsou zkompileovány do DLL knihoven. Hlavní myšlenkou programování v ASP.NET je přenést klasické programování ve Windows na web. Jednotlivé webové prvky jsou poskládány z objektů a přistupuje se k těmto prvkům podobně jako ve Windows. Prvkům lze přiřazovat určité vlastnosti a pracovat s událostmi na nich vytvořených.

Jako software pro internetovou vizualizaci byla vybrána ASP.NET technologie. Konkrétně zdarma dostupný software Microsoft Visual Web Developer 2010 Express.

3.3 Realizace aplikací

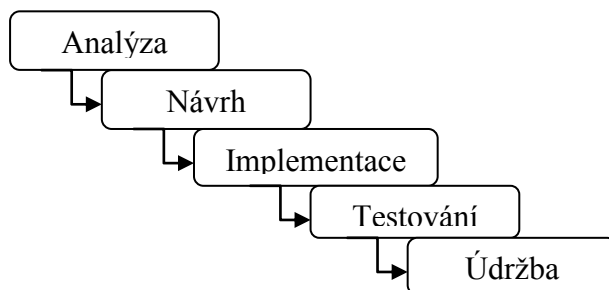
Vzhledem k rozsáhlosti celého vizualizačního a informačního systému je vhodné se držet základních postupů při tvorbě (realizaci) celé aplikace. Bylo využito vodopádového modelu, který vychází z Obr. 3. 7. Dosavadní měřicí systémy jsou navrženy odděleně, z čehož vyplývá určitá nesourodost obou systémů.

Prvním úkolem při realizaci jednoho uceleného vizualizačního a informačního systému je analýza možností. Jak bylo v předešlém textu popisováno, byl zvolen OPC standard, a proto následná analýza směřovala k dalšímu požadavku, který přeposílá data z OPC do databáze. Zde je zvolen software DataHub, který s databází umí komunikovat, kapitola 3.1.1. Dále byla analyzována možná struktura dat v databázi, která má velký vliv na velikost jednotlivých tabulek, viz kapitola 3.1.3. Byly také rozebrány jednotlivé možnosti přístupu jednotlivých aplikací k tabulkám (např. Promotic specifikuje potřebu určité struktury v databázi). Všechny tyto aspekty, byly základem pro správné ukládání dat v čase.

Z výše popisovaného vychází, že základní analýza a návrh ukládání dat do databáze má společnou cestu začínající v DataHubu, který má na jedné straně dva vstupy a na druhé jeden výstup, který končí v databázi, viz Obr. 3.6. Protože struktura obou měřicích celků je v databázi stejně uložená, je jejich následující vizualizace možná přes jeden softwarový prostředek, který bude k oběma celkům přistupovat velmi podobně. Dosavadní analýza a návrh jsou vytvořeny pro část, kde se ukládají data do databáze. Protože tento proces má za úkol sbírat data a správné je uložit, je chápán jako jedna část celého vizualizačního systému.

Druhou a třetí částí je potom přístup vizualizačních aplikací k databázi. Protože je na výběr, jaká technologie bude zvolena pro vyčítání dat z databáze, využívá se zde také analýzy a následného návrhu. Pro tyto potřeby je možné zvolit několik prostředků, které budou plnit funkci lokální nebo internetové vizualizace. Více v kapitolách 4. a 5., které se věnuje návrhu a implementaci. Šestá kapitola rozebírá testování vytvořených aplikací a údržbu celého systému.

Obr. 3.7 Vodopádový model



4. Návrh a realizace lokální vizualizační aplikace

4.1 Návrh

Lokální vizualizace má sloužit pro interní správu. Nabízí se k použití několik programů. Velký důraz při výběru SCADA systému byl kladen na přístupy k informacím do databáze a také byla brána v potaz zkušenost s některými vizualizačními systémy. Nakonec je zvolen SCADA software od Promoticu, který tyto požadavky splňuje. Na Obr. 3.6 je znázorněno jak Promotic přistupuje k aktuálním datům přes OPC a také k databázi, kde využívá SQL příkazů.

Při popisu zobrazování dat přicházejících z OPC serveru je nutné data upravit. Stejný princip úpravy je nutný u všech ostatních vizualizací, protože databáze neobsahuje data přímo k interpretaci. Zobrazování dat je vhodné z pohledu VVP a MVP. Jinak řečeno, je vhodné data zobrazovat zvlášť.

Další náhled je do historických záznamů, kde Promotic nahlíží do MS SQL databáze. Bližší specifikace konečného zhotovení je popsána v kapitole 4.2.

Mimo aplikaci Promotic je vytvořena ještě jedna vizualizace, která je mnohem jednodušší díky tomu, že je založena na protokolu DDE. DataHub poskytuje tento protokol a Word taktéž. Jednoduchým přetažením (funkce Drag and Drop) každé proměnné do Wordu je spuštěna komunikace mezi těmito prostředky.

4.2 Realizace

V první řadě jsou do Promoticu napojena data z obou OPC serveru přidáním objektu PmOPCClienta. Ještě předtím je nainstalován a konfigurován OPC server, viz Příloha II. Kolik je OPC serveru, tolik je nutné nadefinovat přistupujících klientů. V tomto případě jsou nadefinovány dva. Všechny další popisované procedury jsou vytvořeny dvakrát a to vždy pro VVP a MVP. Přidáním objektu PmOPCClientData se nadefinují jednotlivé proměnné. Na Nové aule je v provozu aplikace, která data z VVP zobrazuje. Bylo tedy využito této aplikace k importování potřebných dat do OPC konfigurace. Data z MVP jsou nadefinovaná ručně. Nyní jsou nadefinovány přístupy k datům. Nestačí to však k plnohodnotné vizualizaci.

Každá z tabulek vytvořených v grafickém prostředí reprezentuje jeden vrt. Objekt PmTimer nazvaný „casovac“ pro MVP a „casovac2“ pro VVP, slouží k rozpořívování celé této části vizualizace. Je zde napsán skript (Promotic podporuje Visual Basic), který se spouští

každou vteřinu a provádí příslušné operace. Vyčítá data z OPC a provede vydělení číslem deset pro správnou reprezentaci teplotních dat z MVP. Další vlastnost každého údaje je rozšířená o vizuální znázornění změny (pohybu) teploty. Načtený údaj se uloží a porovná se s údajem jdoucím vteřinu poté. Pokud tento údaj není stejný, zvýrazní se červeně. Každá kontrola probíhá vždy k jednomu předešlému údaji.

V Promoticu je možnost absolutního a relativního adresování. V tomto případě je využito absolutního adresování a vždy je nadefinovaná přímá cesta k dané proměnné. Nyní je popsán tento princip konkrétně pro veličinu A_10 v objektu PmTimer nazvaný „casovac“, který se spouští každou vteřinu. Slovem „dim“ se definují proměnné. A_10 je hodnota přicházející z OPC serveru. Nastaví se tedy cesta k této veličině („Set“). Hodnoty PC_A_10 a a1 jsou už nadefinované v aplikaci v PmData objektu. Samotný kód začíná podmínkou IF, který porovnává hodnotu PC_A_10 s hodnotou A_10/10. V prvním kroku od spuštění v hodnotě PC_A_10 není uložena žádná hodnota, takže se nastaví „a1.Value“ na hodnotu -1. Tím se také údaj ve vizualizaci zobrazí na vteřinu červeně. Po skončení porovnávání se hodnota z A_10 přiřadí do PC_A_10. V další vteřině, pokud se údaj z A_10 nezměnil, nedochází ke zvýraznění a tedy ani hodnota se nemění.

```
dim A_10
Set A_10= pMe.Pm("/vizualizace/OpcClient/DataOPC/#vars/A_10")
dim PC_A_10
Set PC_A_10= pMe.Pm("/vizualizace/DataPC/#vars/PC_A_10")
dim a1
Set a1= pMe.Pm("/vizualizace/DataPC/#vars/a1")

if PC_A_10.Value = (A_10.Value/10) then
a1.Value=1
else
a1.Value=-1
end if
PC_A_10.Value =A_10.Value/10
```

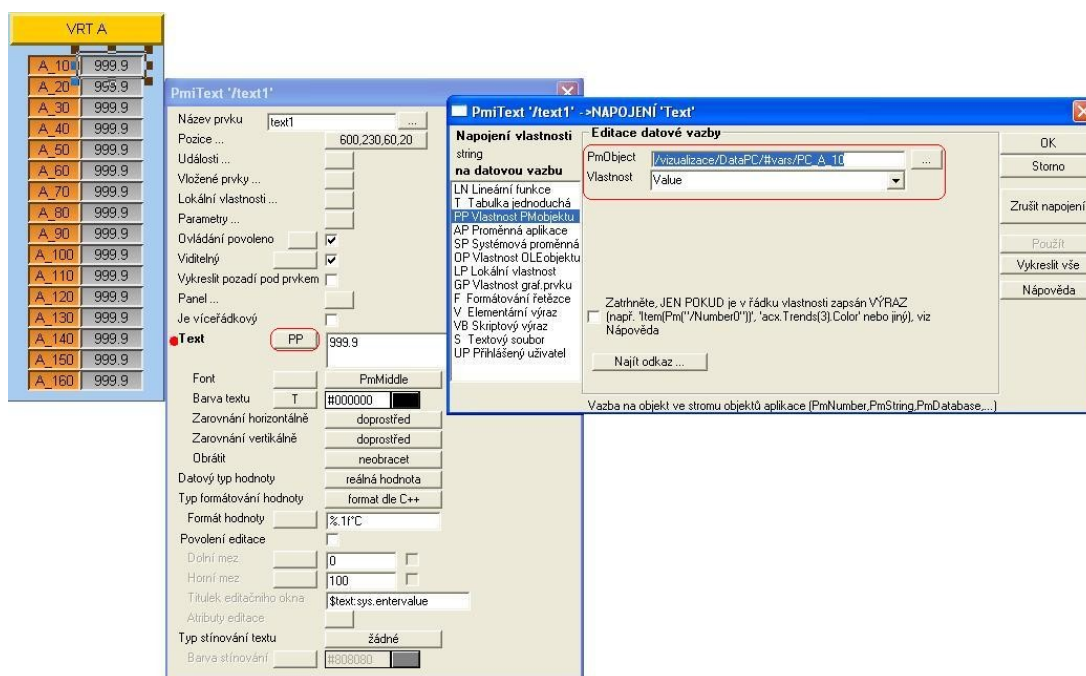
Jednu zobrazovanou hodnotu tedy obsluhují tři proměnné.

Poslední definice je v editoru obrazů, kde se nadefinuje cesta ke každé buňce. Při kliknutí na prvek (buňku) se zobrazí okno znázorněné na Obr. 4.1. Obrázek znázorňuje způsob propojení pod tlačítkem „PP“. Zde se nachází cesta, samozřejmě stejná jako ve skriptu, kde je její vlastnost „Value“. Tímto jednoduchým přiřazením je hodnota nadefinovaná. Definování podpůrné funkce, zvýraznění hodnoty, je na Obr. 4.2, kde pod tlačítkem T (barva textu) a poté pod řídicí proměnnou „PP“ se přiřadí hodnota a1. V poslední řadě se zvolí barva. V aplikaci je při pohybu veličiny zvolená červená.

Co se týče OPC vizualizace, celá výše popsaná procedura je nutná pro všechny ostatní veličiny. Individuální (zdlouhavý) přístup byl zvolen z důvodu několika vrtů, kdy tyto vrty nemají stejný počet teplotních údajů a ani podobné názvosloví. Tímto způsobem je zajištěna přehlednost v celé aplikaci.

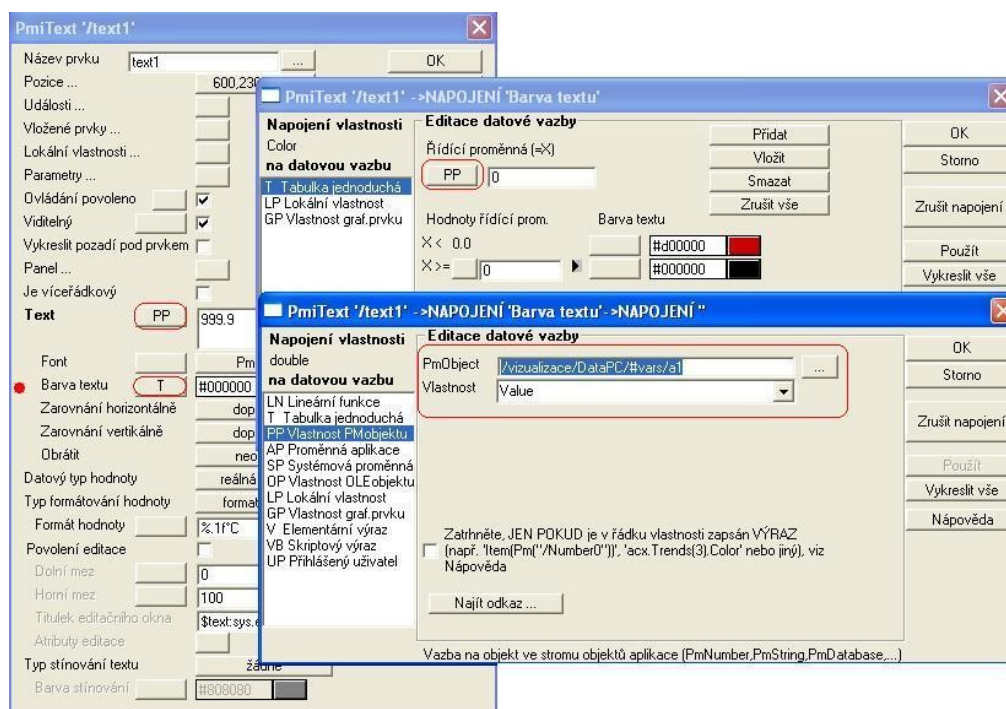
Je oddělen náhled na VVP a MVP. Při spuštění aplikace se zobrazí okno MVP. Toto okno bylo vytvořeno jako první, proto defaultní nastavení je automaticky nastaveno na „Otevřít při startu aplikace“. Na levé straně je menu, které umožňuje přepínání oken. Do okna VVP se aplikace dostane při stisku tlačítka VVP, která vykoná zde vložený příkaz při události „onButtonDown“. Každý zásah (změna) v editoru obrazu je nutné posléze přeložit, aby Promotic mohl tyto grafické obrazy použít.

Obr. 4.1 Nastavení grafického prvku PC_A_10



Zdroj: vlastní

Obr. 4.2 Nastavení pomocné proměnné a1



Zdroj: vlastní

Přístup do databáze byl v první myšlence podobný jako u vizualizace webové. Stiskem tlačítka se měla zobrazit tabulka a graf. Jenomže tento způsob narazil na možnosti Promotivu. Trendviewer lze snadno propojit s PmTrend objektem, který má definovaný přístup k tabulce v databázi, ale se skriptem, který obsluhuje SQL příkazy, je to problematické. Proto byl zvolen způsob, který zvlášť nahlíží na tabulku a zvlášť na graf. Na rozdíl od OPC vizualizace je v jednom okně možné zobrazit vše, tedy tabulky z MVP i VVP.

Tabulkový náhled do databáze je vhodné popisovat z editačního okna. Je zde vytvořeno nové okno s názvem „Panel1“. V editačním okně tohoto panelu se v horní části nachází ComboBox (CB) a prvek ButtonRadio (BR). CB prvek slouží ke zvolení jednotlivých vrtů (tabulek) a zapisuje do objektu PmNumber, který je nazvaný „tabulka“. RB slouží ke zvolení SQL záznamu a zapisuje do stejného objektu nazvaného SQLValue. Do pole „Volby“ jsou zapisována jednotlivá číselná přiřazení, viz Obr. 4.3.

V editačním okně celé aplikace v záložce „Metody“ je vytvořena vlastní metoda nazvaná „LoadFromDBSQL“, která podle volby zobrazuje v tabulce jednotlivé vrty. DbQuery je objekt PmDatabase, přes který jsou jednotlivé SQL příkazy odesílány databázi. Určitá kombinace mezi CB a BR určí, jaký z SQL příkazů se provede. Databáze naplní virtuální

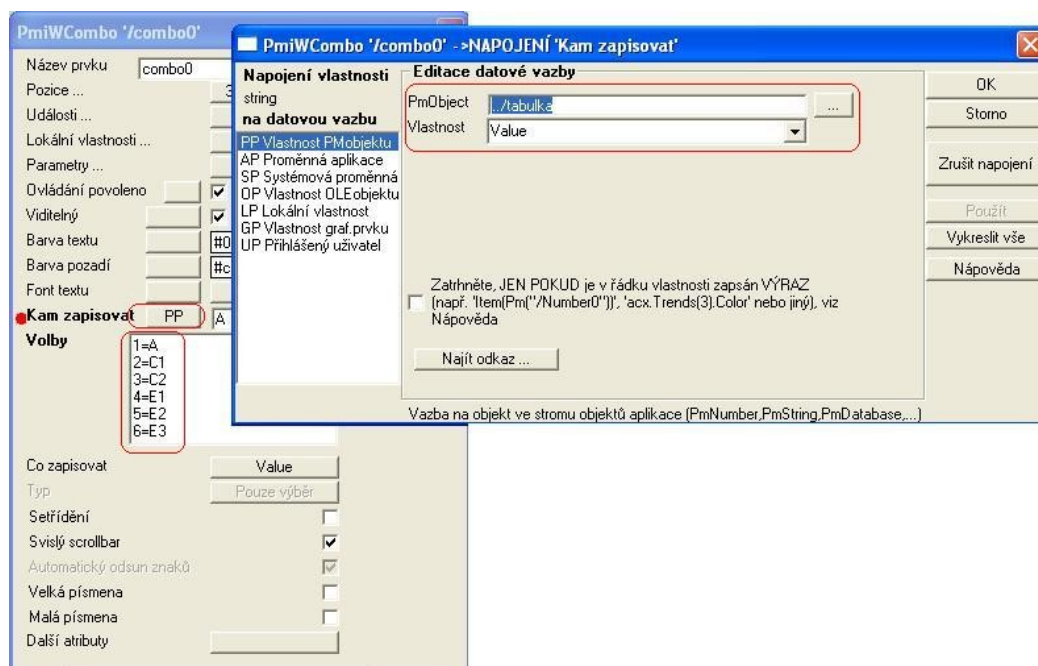
tabulku objektu DbQuery. V grafickém prostředí je potom ActiveX prvek (tabulka), do které se data nahrají.

Níže je znázorněna část skriptu, která obsluhuje vrt A (první vrt) s počtem záznamů za 24 hodin. Při změně požadavku, kdy uživatel zvolí jiný vrt, se tabulka dynamicky překreslí. SQL příkaz má v požadavku „DESC“, což znamená, že data budou zobrazována od nejnovějšího požadavku ke staršímu. Lze jednoduše zkontrolovat, zda celý měřicí systém pracuje a zda jsou data správně ukládána. Je zase nutné dodat, že i v tomto případě jsou zde surová data a uživatel aplikace si tuto skutečnost musí uvědomit, a proto je na ní upozorněn ve vizualizaci.

```
Set oDbQ = Pm("/vizualizace/DbQuery")
nSql = Pm("/vizualizace/SQLValue").Value
tabulka = Pm("/vizualizace/tabulka").Value

if tabulka = 1 then
if nSQL = 0 Then
oDbQ.OpenSQL = "SELECT top 288 * FROM A ORDER BY čas DESC"
```

Obr. 4.3 Editace ComboBoxu

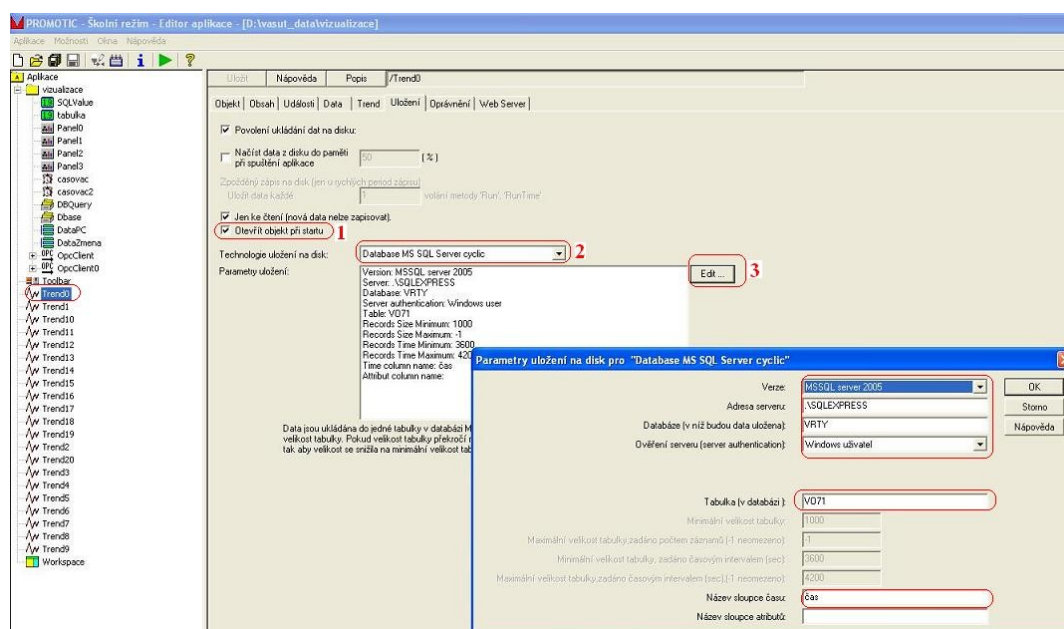


Zdroj: vlastní

Druhou částí je zobrazení grafů. Platí, že data z databáze jsou vyčítána podle názvu tabulky, ale díky objektu PmTrend je možné zobrazovat v jednom grafu více veličin z různých tabulek. Pro grafické zobrazení je použit další ActiveX objekt, zvaný Trendviewer. Jeho

vlastností je přístup k objektům PmTrend, které jsou definované pro všechny tabulky. Konfigurace je provedena v editoru aplikace, kde je nadefinován přístup do databáze. Prakticky jde vždy o stejnou definici s rozdílem vybrané tabulky a vždy zatrženým polem „Jen ke čtení“, viz Obr. 4.4.

Obr. 4.4 Nastavení přístupu objektu Pm Trend do databáze



Zdroj: vlastní

Nastavení je dále provedeno výběrem druhého a třetího bodu. V editačním okně je nastaven přístup k databázi. I přesto, že se používá SQL Server 2008, je konfigurace nastavena na verzi 2005. V poslední řadě je nejdůležitější nastavit název tabulky, který se vždy shoduje s názvem vrtu. Poslední údaj je název prvního sloupce v tabulce, tj. „čas“.

Trendviewer a jeho přístup k datům má předem nakonfigurované prostředky k zobrazování jednotlivých trendů. Tlačítkem v grafu „Načíst“ je na výběr z několika předdefinovaných tabulek, které kopírují tabulky objektů v PmTrend. Avšak nemusí tomu tak být. Trendviewer je modulární. Jednoduchou změnou konfigurace je možné např. zobrazovat data ze všech vrtů v určité hloubce. Záleží jen na uživateli, který způsob je pro něj vhodný. Např. uživatel bude chtít sledovat rozdíly vstupních a výstupních teplot v každém vrtu, aby věděl, zda čerpaná energie má nějaký zisk. Podle Tab. 4.1 lze vidět, jak je možné Trendviewer nastavit. Jsou znázorněny čtyři vrtu, kde je u každého šest teplot. Jednotlivá čidla ve vrtech jsou očíslována, číslo jedna černou (vstup) a červenou barvou (výstup). Číslo dvě označuje následující vrt se stejným barevným rozlišením jako číslo jedna, atd. Objekt PmTrend v tomto

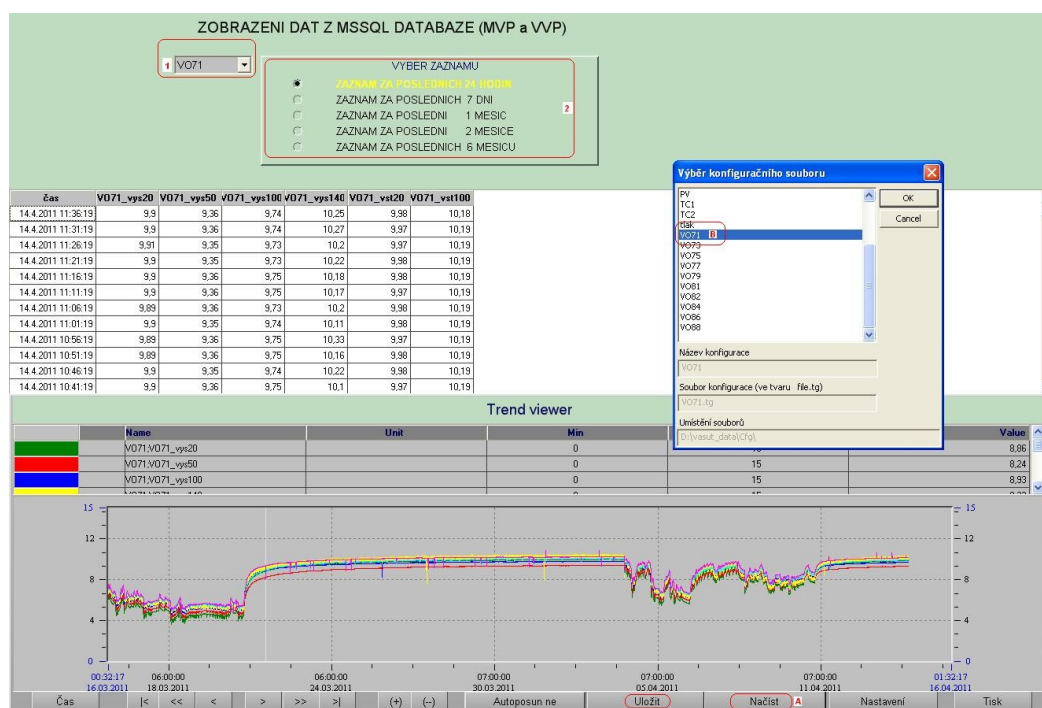
případě nahraje všechny čtyři tabulky do své aplikace a podle nastavení Trendvieweru vybere jen ty záznamy, které jsou nakonfigurovány.

Tab. 4.1 Možnosti vyčítání v Trendvieweru

	reprezentace	čas	20m	50m	100m	140m	20m	100m
Trend0	VO71		1				1	
Trend1	VO72		2				2	
Trend2	VO73		3				3	
atd.	VO..		4				4	

Zdroj: vlastní

Obr. 4.5 Grafická a tabulková vizualizace



Zdroj: vlastní

Pro sledování jednotlivých tabulek, jsou přednastaveny všechny teplotní vrty s příslušnými hodnotami z obou polygonů. Práce s tabulkou je velmi rychlá a jednoduchá. Stačí zvolit jméno vrtu v ComboBoxu a poté výběr záznamu. Grafický náhled je mírně složitější.

V grafickém náhledu jsou přednastavené konfigurace teplotních vrtů. Stiskem tlačítka (A) „Načíst“ se zobrazí okno „Výběr konfiguračního souboru“ (B). Po zvolení se začnou načítat data a zobrazí se v grafu. Tento proces může být zdlouhavý, pokud je časový rozsah velký,

protože dochází k vyčtení celé tabulky. Aby byly vidět teplotní průběhy, je nutné zmenšit časové měřítko tlačítkem „(-)“. Při každé této akci se data dočítají a nabývají na úplnosti. Při hledání některých údajů jsou k dispozici další navigační tlačítka. Tlačítko „čas“ slouží k zobrazení časového údaje. Pokud je hledán určitý záznam v daném čase, zobrazí se na pravém okraji grafu. Po načtení tabulky v grafu je možné tyto data rovnou analyzovat. Stiskem na určité místo se zobrazí vertikální čára a v horní části Trendvieweru se zobrazí hodnoty v místě označení. Bylo také zmíněno, že je možnost dalších variabilních nastavení. To se provádí v tlačítku „Nastavení“, kde se zvolí trend a jeho příslušné hodnoty. Po nakonfigurování je vhodné data uložit. Jedná se o soubor s příponou „tg“, který Promotic umí obsloužit. Je možné si vytvořit vlastní konfiguraci, viz Příloha VII.

5. Návrh a realizace internetové vizualizační aplikace

5.1 Návrh

V rámci návrhu internetové vizualizace je třeba zajistit strukturovaný náhled zpracovaných dat na internetu. Snadná manipulace a vizuální zobrazení ve formě tabulek a hlavně grafu by neměla chybět. Dále je třeba zobrazovat informace o měřených veličinách, obrázky fyzicky rozložených polygonů (jejich vrtů) a další psané informace týkající se této problematiky. Přínos bude mít i rubrika o aktuálních novinkách, které se nemusí týkat jen zásahu do webové vizualizace. Další požadavek je kladen na uspořádání celé vizualizace. Pro přehlednost je vhodné vytvořit určité zóny, které budou reprezentovat určité funkce. Např. menu bude na levé straně obrazovky a po celou dobu přepínání na jiné stránky by tomu nemělo být jinak. Všechny zmíněné požadavky povedou k snadné orientaci a informovanosti pro uživatele.

Většina těchto stránek bude statická. Opakem budou stránky VVP a MVP, kde bude zobrazování dat nejsložitějším úkolem. VVP a MVP budou zobrazovány rozděleně. Uživatel, který potřebuje zobrazení dat např. z MVP, bude hledat z nabídky vrtů z MVP. Přispěje to taktéž k jednodušší orientaci. Při zobrazení internetové stránky MVP se zobrazí nabídka vrtů, kde si jeden vrt uživatel vybere. Správně navržené tabulky, jeden vrt odpovídá jedné tabulce, napomáhají k přehlednosti a tedy i k jednodušší reprezentaci. Při zvolení určitého vrtu se dále zvolí čas zobrazení potřebných záznamů ve smyslu zadání začátku a konce. Posléze, po potvrzení, se zobrazí graf s daty a tabulka. V principu jednoduché operace pro uživatele, budou v pozadí ukrývat vyčítání dat z MS SQL databáze. V případě MVP je třeba podělení všech hodnot určitým číslem pro správnou reprezentaci záznamů, aby byly zobrazovány reálné hodnoty. VVP obsahuje reálné hodnoty rovnou. Zde bude třeba přizpůsobit kód pro zaokrouhlení na určitá desetinná místa. Po provedení zvoleného procesu bude možno zvolit jiný časový úsek i změnu jiné tabulky.

Další podstatnou funkcí by mělo být stažení těchto nadefinovaných požadavků v určitém formátu, např. v CSV souboru. Po zvolení časového údaje (hodin, dnů, měsíců) a po vybrání vrtu se zobrazí vygenerovaný soubor ke stažení, který nabídne možnost práce s daty např. v Excelu.

Další požadavky jsou z hlediska technických možností. Internetová vizualizace, její návrh a následná realizace je možná provádět v různých softwarových prostředcích. Záleží, která z technologií je programátorovi bližší.

Pro tvorbu této vizualizace byla zvolena technologie ASP.NET, která má určité výhody. Především OOP (Objektově orientované programování). V tomto druhu programování se

využívají objekty, ke kterým programátor přistupuje přes rozhraní. Aby objekty a celkově OOP mělo smysl, všechny objekty jsou zapouzdřeny. Je dovoleno přistupovat pouze přes rozhraní, aby nedocházelo k nekonzistenci těchto prvků. Jinak řečeno, programátor si zvolí určitý objekt, který má navenek dělat určitou operaci a nestará se o vnitřní záležitosti objektu. Zajímá jej pouze rozhraní a předávané parametry.

Při návrhu internetové vizualizace bylo důležité zjistit, jaké objekty jsou k dispozici. Spíše jde o opak, tj. jaké nejsou, protože tato technologie je přímo určená pro tvorbu webových stránek a její možnosti jsou rozsáhlé i bez doinstalování dalších podpor.

5.2 Realizace

5.2.1 Programová část

V kapitole Návrh byly rozebrány požadavky na zobrazení. Následující řádky této kapitoly popisují konečnou funkční webovou aplikaci z hlediska programu.

Pro realizaci webové vizualizace se zvolila technologie ASP.NET a zvolený programovací jazyk je C#, v programové verzi Microsoft Visual Studio 2010 Express v odnoži Microsoft Visual Web Developer 2010 Express.

„Základní kámen“ celé struktury zobrazování je hlavní vzhled stránek, který je vytvořen v šabloně zvané Master Page (MP). Hlavička, tělo a patička jsou tedy záležitostmi šablony, která se bude objevovat na všech stránkách. Je zde nadefinováno rozvržení stránky a nechybí ani dominanta stránky (titulní obrázek). MP je rozeznatelný direktivitou `<%@ Master%>`. `MasterPage.master` je odkaz na MP šablonu sloužící ostatním stránkám. Webové podstránky potom neobsahují hlavičky, těla a ani patičky, ale obsahují odkaz na `MasterPage.master`. Ve výsledku se potřebná stránka sloučí s MP stránkou a na venek tvoří jeden celek. Dochází tak k jednodušším a rychlejším návrhům, nemluvě o tom, že stejná struktura základního zobrazení je na místě.

Dále je do MP stránky vložen `ContentPlaceholder` nazvaný „main“, který slouží pro zobrazení obsahu ostatních stránek. Samozřejmě je možnost vytvořit více `ContentPlaceholderu`, potom je třeba dbát na odkazy ve vytvořených stránkách, například takto vypadá odkaz na MP stránku stránky `Default.aspx`.

```
<asp:Content ID="Content1" ContentPlaceHolderID="main" Runat="Server">
```

Takovýto zápis obsahuje každá stránka a má přesně zapsané ID Holderu.

Další podporou při navrhování webových stránek jsou CSS kaskádové styly. V zásadě jde o rozvržení pozic jednotlivých zobrazovaných prvků (tabulek, hlavního menu a obrázků). V rámci MP stránky jsou vytvořené tři základní tabulky, které oddělují obrázek hlavní stránky, menu s výběrem a obsahovou tabulku. ContestPlaceholder je poté logicky vložen do obsahové tabulky, která se bude dynamicky měnit změnou přístupu na jinou stránku. Při použití Drag and Drop funkce (přetažení prvků do webového obsahu) se nadefinují pouze základní styly každého prvku. V programovém prostředí při přechodu na záložku „Source“ se můžou tyto styly předefinovat. To platí hlavně při nastavení zarovnání, např. hlavní obrázek by měl být uprostřed a menu na levé straně (platí v rámci vytvořené tabulky).

Hlavní menu s polohou na levé straně slouží pro navigaci uživatele na webu. Zde je využit Treeview objekt a SiteMapDataSource [10]. Tyto dva objekty postačí pro funkční navigaci mezi stránkami. Treeview zobrazuje fyzické názvy stránek a přesměrovává je na jiné stránky. Dále využívá zdroje dat SiteMapDataSource (dále SMDS), který má nadefinované propojení. V okně Solution Explorer se pomocí SMDS vytvořil konfigurační soubor (zdroj dat) pro Treeview. Defaultně má vytvořené nějaké stránky pro informaci programátora. Tento konfigurační soubor je pro informaci částečně zobrazen níže.

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap>
  <siteMapNode title="Home" description="Home" url="~/default.aspx">
    <siteMapNode title="Novinky" description="novinky"
      url="~/novinky.aspx" />
    <siteMapNode title="Vizualizace" description="Vizualizace obou polygonů"
      url="~/vizualizace.aspx" />
    <siteMapNode title="Malý výzkumný polygon" description="Malý výzkumný polygon"
      url="~/MVP.aspx" />
    <siteMapNode title="Velký výzkumný polygon" description="Velký výzkumný polygon"
      url="~/VVP.aspx" />
  </siteMapNode>
  <siteMapNode title="katedra 450" description="katedra 450"
    url="http://dmc.vsb.cz/" />
</siteMapNode>
</siteMap>
```

Každá stránka obsahuje název a příslušný odkaz. Stránky komunikující v rámci webové aplikace začínají znakem (~).

Teprve nyní jsou nadefinované všechny potřebné funkce hlavní (MP) stránky. Vytvoření nové webové stránky je možné v okně Solution Explorer – kliknutím pravým tlačítkem myši na odkaz projektu (začínajícím C:\) nebo klikem na Ctrl+Shift+A a výběrem Web Form. Po vygenerování a pojmenování stránky, je ji nutné zakomponovat do MP stránky

zápisem `MasterPageFile=~/MasterPage.master` a zápisem `HolderID`. Tento kód vypadá následovně:

```
<%@ Page Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true"
CodeFile="ault2.aspx.cs" Inherits="ault2" %>
<asp:Content ID="Content1" ContentPlaceHolderID="main" Runat="Server">
    <p>
        zde se píše text</p>
</asp:Content>
```

Je nezbytné vymazat předešlou strukturu, která zde obsahuje hlavičku, tělo a patičku, protože aplikace v základu předpokládá vytvoření samostatné webové stránky. V záložce Design se vkládá text a to do oblasti Placeholderu, který byl v CSS stylu nadefinován. Tato procedura je u ostatních stránek stejná. Jednoduchým zkopírováním bylo vytvořeno několik potřebných stránek. Podle výše popsané procedury je tvorba statických stránek jednoznačná. Vkládání textů a obrázků podléhá obdobnému strukturování jako ve Wordu. Jenom nahrání obrázků do projektu je komplikovanější. Je nutné daný obrázek dodat do složky webové aplikace. Funkcí Drag nad drop se do Designu stránky vloží ze Solution Exploreru, kde by měl být vidět. Následně je nutné zvolit cestu k souboru. Všechny obrázky jsou ve formátu JPEG, protože svojí fyzickou velikostí jsou předurčeny na web.

Dynamická část vizualizace se nachází na stránkách VVP a MVP. Kódy jsou na obou stránkách velmi podobné (přístup k databázi a zpracování surových dat). Rozdíly jsou v konečné interpretaci výsledků. Co se týče základního zobrazení, platí stejné uplatnění MP stránky společně s kaskádovými styly. Dále bude popsán jeden z polygonů, např. MVP.

V kódu MVP je využito příkazu switch, který přepíná obsah kódu pro vykonání podle zvoleného požadavku v DropDownListu. Proto je další popisování soustředěno na jeden případ vybraného vrtu. Tento způsob byl zvolen z důvodu individuálního přístupu pro každý vrt. Dále má obrovskou výhodu v přehlednosti kódu. V rámci zvolené tabulky se pokaždé zobrazuje jiný počet řádků a přitom je nutná konečná úprava (podělení nebo vynásobení o určitou hodnotu). Přes třicet tabulek o několika kolonkách má vlastní kód. V rámci každého kódu je nutné definovat graf pro přesný počet záznamů. Dále je nutné těmto záznamům definovat barvu a možnost dočasného zakrytí v grafu k potřebné analýze. CheckBoxy jsou defaultně nastaveny jako „checked“, tzn., že při vygenerování grafu jsou všechny záznamy graficky zobrazeny. Jejich viditelnost je však povolena jenom při vygenerování požadavku. Při prvotním spuštění aplikace vidět nejsou.

Začátek C# kódu v MVP.aspx.cs začíná definováním použitých tříd „using System“ a dalších odvozených tříd. Dále je vytvořena třída MVP, která dědí vlastnosti třídy Systém.Web.UI.Page. Díky tomu je možnost využití těchto vlastností na webu.

Metoda Button1_Click odstartuje kontrolu kalendářů. Nachází se zde podmínky, které ošetřují nastavený čas. Pokud do kalendáře uživatel nezadá hodnoty, nastaví se dnešní datum s časem 00:00:00 a konečný den s časem 23:59:59. Tato část kódu s popisky je uvedena níže.

```
if (Calendar1.SelectedDate != DateTime.Parse("1.1.001 0:00:00"))
    //když kalendář není roven ("1.1.001 0:00:00"), poté je použito datum z kalendáře
{
    date1 = Calendar1.SelectedDate;
}
else
{
    date1 = Calendar1.TodaysDate;           //vložení dnešního dne s časem 00:00:00
}
if (Calendar2.SelectedDate != DateTime.Parse("1.1.001 0:00:00"))
{
    date2 = Calendar2.SelectedDate.AddHours(23).AddMinutes(59).AddSeconds(59);
    //pouze přidá tento čas ke zvolenému datu (23:59:59)
}
else
{
    date2 = Calendar2.TodaysDate;           //vložení celého údaje
    date2 = Calendar2.TodaysDate.AddHours(23).AddMinutes(59).AddSeconds(59);
}
```

Následující první řádek zobrazuje zápis vytvořené instance „cteni“, který obsahuje odkaz na SQL_pripojeni. Jedná se o text, který obsahuje cestu k SQL serveru. SQL_pripojeni se v projektu nachází na prvních řádcích programu. Ukazatel je poté zapsán ve web.config a jeho znění je uvedeno ve třetím bodě.

- `SqlConnection cteni = new SqlConnection(SQL_pripojeni);`
- `string SQL_pripojeni = WebConfigurationManager.ConnectionStrings["SQL_Spojeni"].ConnectionString;`
`// prevzeti SQL_Spojeni z web.config`
- `<add name="SQL_Spojeni" connectionString="Data Source=VEC-VRTY\SQLEXPRESS;Initial Catalog=VRTY;Integrated Security=True" providerName="System.Data.SqlClient"/>`

Nyní se program dostává do oblasti se switchem, kde se podle zvoleného vrtu provede potřebný proces. Při představě, že uživatel zvolil např. první vrt, program přejde na úroveň case 0. Jako první se provede SqlCommand, který použije vložený string a propojení společně s vloženými časovými parametry. Metodou cteni.Open() se provede SqlCommand.

Následuje reader, který si uloží vyčtená data do streamu. Stream není nijak strukturovaný, proto byla vytvořena tabulka nazvaná „dt“, do které se data musí strukturovat. Prvně je však důležité přidání názvů sloupců, které se definují příkazem dt.Columns.Add. Jako první je pojmenován sloupec „čas“, následuje „A_10m“ atd. Takto je nadefinováno tolik sloupců, kolik je potřeba.

case 0:

```
SqlCommand cmd = new SqlCommand("SELECT * FROM A WHERE čas>=@time1 and čas<=@time2 order by čas",  
cteni); //vložení SQL příkazu a vložení propojení (čtení)
```

```
cmd.Parameters.Add("@time1", SqlDbType.DateTime).Value = date1; //vložení časového údaje do stringu
```

```
cmd.Parameters.Add("@time2", SqlDbType.DateTime).Value = date2; //vložení časového údaje do stringu
```

```
cteni.Open(); //vložení časového údaje do stringu
```

```
SqlDataReader reader = cmd.ExecuteReader(); //načtení dat do streamu
```

```
DataTable dt = new DataTable(); //vytvoření virtuální tabulky
```

```
dt.Columns.Add("čas", Type.GetType("System.String")); //definování názvu kolonek
```

```
dt.Columns.Add("A_10m", Type.GetType("System.Double")); //definování názvu kolonek
```

..... a další.

Prázdnou tabulku je teď nutné naplnit ze streamu. K tomu slouží níže vytvořená procedura postupného naplňování. Jestli je na webové stránce označen CheckBox38, uživatel zvolil hodinový výběr (při startu aplikace je tento CheckBox vždy zatržen). Spustí se „reader“. V následující podmínce je tento zápis ($c \% 12 == 0$). Číslo dvanáct udává počet záznamů v databázi za hodinu. Do tabulky „dt“ je vložen každý dvanáctý záznam, jehož zbytek po vydělení je vždy nula. Tím je jednoduše vyřešena procedura zobrazení hodinových záznamů. V rámci podmínky „while“ při splnění podmínce po celočíselném zbytku se přidávají řádky. Přidá se čas uložený ve stringu a posléze hodnota, která je upravena do konečného tvaru.

```

if (CheckBox38.Checked)           //jestli je checkbox.checked provede se následující.
{
    int c = 0;
    while (reader.Read())
    {
        if (c % 12 == 0)           //zbytek po celočíselném dělení == 0, provede se
        {
            dt.Rows.Add();          //přidání řádku
            dt.Rows[dt.Rows.Count - 1][0] = reader[0].ToString(); // přidání časového údaje beze změny

            for (int i = 1; i < reader.FieldCount; i++) // přidání hodnot s následným podělením
            {
                int n = (int)reader[i];
                dt.Rows[dt.Rows.Count - 1][i] = (double)(n / 10f);
            }
        }
        c++;
    }
}
else

```

Pokud výše zmíněný CheckBox nebude označen, tato podmínka není splněna a za „else“ následuje stejná procedura, která však neobsahuje podmínku *if (c % 12 == 0)*. Tím pádem se do tabulky přidají všechny hodnoty ze streamu. Ve výsledku tabulka obsahuje stejný počet záznamů jako databáze s ohledem na počáteční a konečný požadavek na čas. Poslední kroky programu je přiřazení tabulky a grafu na datový zdroj „dt“. „Databind()“ metoda vyvolá vazbu na datový zdroj. Poslední nutnou věcí je zavření readru a čtení z QSL serveru.

```

Chart1.DataSource = dt;
Chart1.DataBind();
GridView1.DataSource = dt;
GridView1.DataBind();
reader.Close();
cteni.Close();

```

Doposavad byla popsána procedura, která zobrazuje zmíněný vrt A. Do tohoto popisu nejsou zahrnuty méně podstatné kroky a je soustředěn na stěžejní část vykonání programu. Pro stažení dat do CSV souboru je však nutné popsat další procedury. Zde jsou taktéž zvoleny podmínky switche. Začátek je stejný jako při zobrazování na webu. Aby popis byl kompletní, bude aplikován na vrt A, který byl rozebrán už výše.

Ve switchi se nachází jenom SqlCommandy. Dále následuje SqlDataAdapter, který si obsah SqlCommandu přivlastní. Vytvořený adapter naplní dataset. Nyní se programová část pohybuje pouze v oblasti vnitřní paměti z důvodu následného poskytnutí uživateli.

MemoryStream a StreamWriter slouží k potřebnému strukturování. Níže je zobrazen popis, jak data z datasetu strukturovat do MemoryStreamu.

```

SqlDataAdapter adapter = new SqlDataAdapter(command);           //třída, která představuje rozhraní mezi databází a datasetem
DataSet dataset = new DataSet();                               //vytvořená třída dataset
adapter.Fill(dataset);                                         //naplnění datasetu daty z databáze
MemoryStream s = new MemoryStream();                          //MemoryStream třída ukládá data do mezipaměti
StreamWriter w = new StreamWriter(s);                         //StreamWriter pracuje s proudy dat a zapisuje do s
    
```

Je vytvořena tabulka s názvem „table“. Níže zobrazená část provádí proceduru zápisu dat do jednotlivých řádků. V první řadě příkaz foreach prochází postupně jednotlivé záznamy a nastavuje indexy.

```

int rowIndex = 0;                                             //index řádků
foreach (DataRow row in table.Rows)                          //procedura zápisu dat do
rowIndex++;                                                  //načítání indexu
ColumnIndex = 0;                                             //index pro sloupce
    
```

Druhý příkaz se pohybuje v oblasti jednotlivých řádků, kdy index o počáteční hodnotě „ColumnIndex==1“ (pohyb příkazu v prvním sloupci), zapíše do řádku čas. Pokud však jde o vyšší číslo, provádí se proces ukládání podle vybraného vrtu. Převody teplot, které jsou na prvních osmi pozicích v DropDownList1 jsou vyděleny desíti a následně uloženy do streamu. Nachází se zde konverze z Int na Double, aby čísla mohla být zobrazena s desetinnou hodnotou. Jsou zobrazeny ještě další možné vrtu. Vrt s hodnotou devět a patnáct jsou podělené miliónem pro správnou reprezentaci.

```

foreach (DataColumn col in table.Columns)
ColumnIndex++;
if (ColumnIndex == 1)
{
w.Write(row[col.ColumnName]);                               //zápis času do každého prvního sloupce s daty
}
else
{
if (DropDownList1.SelectedIndex < 9)                       //první teploty do 8 jsou podělené 10
{
w.Write((double)((int)(row[col.ColumnName])) / 10);
}
else if (DropDownList1.SelectedIndex == 9 || DropDownList1.SelectedIndex == 15)
//tlak a Prutok_dalsi děleno 1000000
{
w.Write((double)((int)(row[col.ColumnName])) / 1000000);
}
}
    
```

Tímto končí jedna možná varianta výběru z vrtu A. Kompletní popis jednoho vrtu je totožný s ostatními vrty.

Část druhá, ukládání do CSV souboru, je programově velmi krátká. Protože není třeba dodatečných nastavování jako u grafů a tabulek. Dále zde není zakomponováno hodinové vzorkování hodnot z databáze. Do jisté míry by tato možnost mohla být brána jako nežádoucí z hlediska analyzování v malém rozsahu vzorku. Nicméně existují dva způsoby, jak data analyzovat s potřebným hodinovým vzorkem. Při uložení dat do Excelu je možné využít jeho funkcí a data jednoduše přizpůsobit. Druhá jednodušší možnost spočívá ve vykreslení dat na webu. Vybere se vrt s potřebnými údaji a nechá se vykreslit. V tabulce se v prvním záznamu myši klikne před čas. Stiskne se Shift a druhý klik myši se klikne na konec celé tabulky. Zkopírováním a vložením (Ctrl-C a Ctrl-V) se zajistí zobrazení dat v Excelu.

Je nutné brát v úvahu velikost vybraného záznamu. Např. když uživatel vybere časové rozmezí dvou měsíců a nechá jej vykreslit, zobrazení se projeví delší časovou odezvou (platí pro zobrazení bez hodinového vzorku). Dva měsíce odpovídají přibližně 17 000 záznamů o velikosti cca 2 MB, proto stáhnutí dat do počítače uživatele trvá déle. Naproti tomu uložení tak velkého objemu dat do textového souboru zabírá přibližně 300 kB. Je tedy vhodné objemné data stahovat přes textový soubor a následně je analyzovat na vlastním počítači. Změna časové konstanty u teplot je výrazně vyšší, než u jiných měřených veličin. Proto hodinový způsob zobrazení je vhodný u teplot, kdy jsou změny pomalé. Příkony, průtokoměry aj. mění své hodnoty velmi rychle. Nahlížení na ně je vhodné spíše v rozmezí týdne než měsíce a ještě s vypnutým hodinovým náhledem.

5.2.2 Popis funkce webové vizualizace

Při vstupu na webovou vizualizaci se na levé straně menu nachází odkaz Vizualizace. Zde je návod, jak se správně orientovat ve vyhledávání na MVP a VVP. Stránka Vizualizace pod sebou zahrnuje oba měřicí polygony.

Po příchodu na jeden z těchto polygonů je možno výběru jednotlivého vrtu v Dropdownlistu, na Obr. 5.1 je znázorněný číslem 1. Poté se vybere datum, bod 2 a 3. Pokud je třeba zobrazit záznam do aktuálního času, není třeba druhý kalendář nastavovat. Čtvrtý bod je společný pro tlačítko Start a ikonu Excelu. Po vyvolání události na tlačítku Start se vykreslí graf a tabulka. Událost na ikoně nabídne soubor ke stažení podle výše vybraných požadavků v CSV souboru. Tento soubor se snadno importuje do Excelu, kde lze data analyzovat podle vlastních požadavků. Pátý bod označený „vzorek/hod“ slouží k výběru pouze jednoho záznamu za hodinu. Při načtení stránky, je funkce vždy zapnutá.

Obr. 5.1 Výběr vrtu

vyberte vrt MVP:

1 4

Soubor ke stažení:

4

☒ vzorek/hod 5

2

Zadejte počáteční datum

únor 2011						
po	út	st	čt	pá	so	ne
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	1	2	3	4	5	6
7	8	9	10	11	12	13

3

Zadejte konečné datum

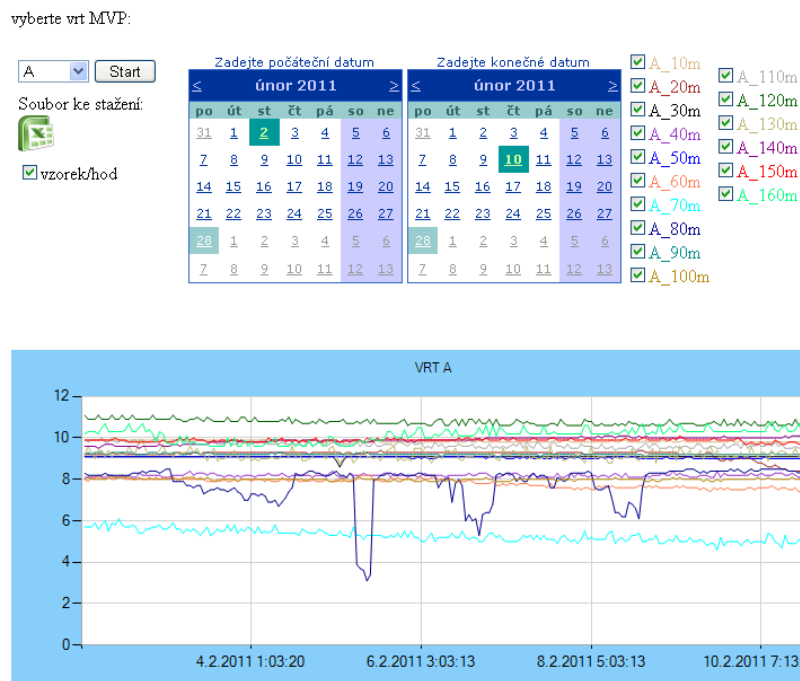
únor 2011						
po	út	st	čt	pá	so	ne
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	1	2	3	4	5	6
7	8	9	10	11	12	13

Zdroj: vlastní

Kalendář

Nastavuje počáteční a konečný čas výběru. Na horní pravé a levé straně objektu jsou šipky, které posouvají kalendář o měsíc dopředu či zpět. Pokud se na jednom z kalendářů nevybere datum, nastaví se aktuální čas.

Obr. 5.2 Událost při zvolení výběru vrtu s hodinovým vzorkem



Zdroj: vlastní

Obr. 5.3 Událost při zvolení výběru vrtu, zobrazená celá databáze

vyberte vrt MVP:

A

Soubor ke stažení:

☐ vzorek/hod

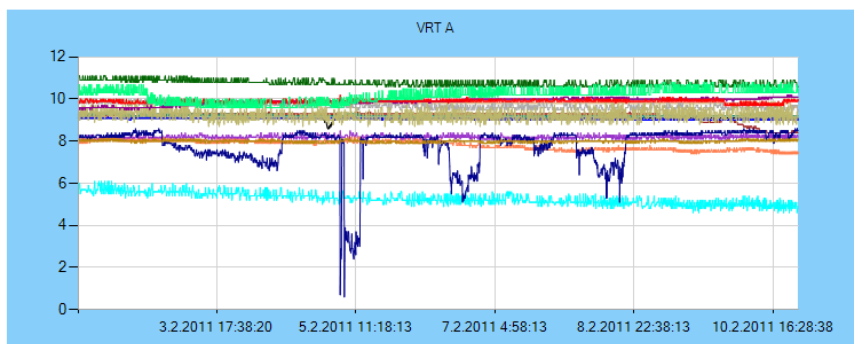
Zadejte počáteční datum

únor 2011						
po	út	st	čt	pá	so	ne
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	1	2	3	4	5	6
7	8	9	10	11	12	13

Zadejte konečné datum

únor 2011						
po	út	st	čt	pá	so	ne
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	1	2	3	4	5	6
7	8	9	10	11	12	13

- ☒ A_10m
- ☒ A_20m
- ☒ A_30m
- ☒ A_40m
- ☒ A_50m
- ☒ A_60m
- ☒ A_70m
- ☒ A_80m
- ☒ A_90m
- ☒ A_100m
- ☒ A_110m
- ☒ A_120m
- ☒ A_130m
- ☒ A_140m
- ☒ A_150m
- ☒ A_160m



Zdroj: vlastní

Po vyvolání události na tlačítku Start se zobrazí graf. Na Obr. 5.2 (s hodinovým vzorkem) je znázorněn vrt A v únoru 2011. Pro porovnání ve stejném časovém úseku Obr. 5.3 zobrazuje všechny záznamy.

Počáteční čas je vždy nastaven na začátek dne. Konečný čas je nastaven na 23:59:59, pokud však není zvolen záznam do současnosti, kde tento čas je čas aktuální. Na pravé straně od kalendáře jsou Checkboxy, které umožňují zapínat a vypínat zobrazení údajů v grafu. Defaultně jsou všechny průběhy zobrazeny. Jednotlivé záznamy jsou rozlišeny přesně nadefinovanými barvami, které nelze měnit. Níže pod grafem je potom zobrazena tabulka, která zobrazuje stejné údaje jako graf.

6. Testování vytvořených aplikací

Testování vytvořených aplikací začíná u DataHubu, který přeposílá hodnoty do databáze. V podstatě jde o kontrolu správné konfigurace dat, které se mají odesílat. Jde hlavně o datové typy, které musí být Real, Int nebo Datetime, podle polygonu. Pokud je tohle nastavení správné a do databáze přibývají data se správným datovým typem, je chybovost takto nastaveného zápisu vyloučena. OPC servery a jejich poskytované data v určitých typech korespondují s datovými typy v DataHubu. DataHub přesně vidí a používá datové typy databáze.

Při pohledu na webovou vizualizaci, kdy aplikace stahuje data z tabulky v databázi, je předpoklad, že první sloupec je typu Datetime a převádí si jej na datový typ String. Další sloupce jsou typu Int nebo Real. Je nutné dbát na správné definování. Tento předpoklad je třeba dodržet a poté bude aplikace fungovat správně. Chyba může nastat při přidávání dalších tabulek, proto je nutná průběžná kontrola. To stejné platí i pro tabulky v databázi, kdy definice tabulky musí korespondovat s požadavky v DataHubu.

Promotic je software, který je léta prověřený praxí. Návrh vizualizační aplikace v Promoticu je záležitost nastavování a určitého programování. Vytvářené skripty a jejich správnost je zkontrolována vlastním debuggem Promoticu. Při vyskytnutí chyby v aplikaci Promotic vizualizaci nespustí nebo není spuštěna korektně. Pro ladění těchto problémů je možné nahlédnout do zvláštního okna zvané „Errors“, ve kterém jsou podrobnosti o nekorektním chování systému. Při ladění aplikace bylo zmíněné možnosti využito. Funkčnost celé aplikace byla prověřena.

Dalším typem testování lze chápat odpověď aplikace na dotaz pod zatížením (neboli vyvoláním umělého přetížení webového serveru). V tomto smyslu bylo ve stejný okamžik spuštěno pět internetových stránek s rozsáhlým požadavkem. Ve výsledku CPU Serveru dosahovalo 60% zatížení po dobu čtyř vteřin. Zatížení databáze bylo vteřinové s přenosem 8MB dat. Databáze je velmi rychlá a následné zpracování internetové vizualizační aplikace trvalo čtyři vteřiny. Lze konstatovat, že taková okamžitá extrémní zátěž nebude častá. Během zkoušení a zatěžování webové aplikace se neobjevili žádné chyby, které by narušovali činnost aplikace.

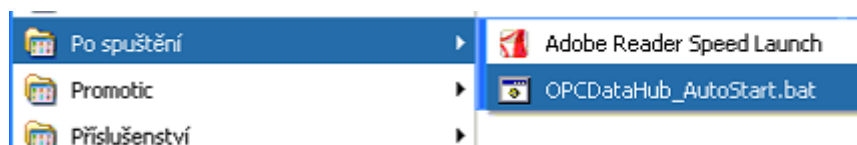
Ať už je databáze zatížená nebo není, její odezvy při požadavku jsou vždy relativně rychlé a bez problémů se žadatelům dostane požadované odpovědi. Tento software, jako každý jiný, má k dispozici aktualizace, které je vhodné občas doplnit.

Co se týče přenosu těchto dat k uživateli, existuje závislost na propustnosti sítě a na rychlosti zpracování na straně uživatele.

Testování z hlediska funkčnosti PC bylo také nutno řešit. Stávalo se, že OS Windows XP se po nainstalování aktualizací sám restartoval. Proto byly vypnuty aktualizace.

Software DataHub ve verzi 6. není kompatibilní s diakritikou české lokalizace Win XP. Nemůže najít cestu k vlastním složkám. Spuštění DataHubu je možné uskutečnit přes příkazovou řádku s přesně zadanou cestou spuštění. Vzhledem k tomu, že při restartu PC je třeba okamžitého naběhnutí Data Loggingu, byl vytvořen BAT soubor, který se o tohle spuštění postará sám a to po nastartování systému Windows. Více informací se nachází v Příloze VI.

Obr. 6.1 Spuštění DataHubu



Zdroj: vlastní

Tento software dále kontroluje přicházející údaje z OPC a zjišťuje, zda jsou hodnoty ve správných mezích. Pokud ne, odešle se správci serveru informační email [9].

7. Zhodnocení výsledků

Při tvorbě této diplomové práce jsem byl postaven před problém, při kterém jsem měl sbírat informace ze dvou různorodých systémů. Bylo třeba obsluhovat přibližně 200 čidel, které přenášejí data, jako jsou z větší části teplotní údaje a potom doplňkové informace označené jako technologické (výkony, průtokoměry atd). Základní požadavek, který bylo nutné splnit, je ukládání v pětiminutových časových intervalech.

Využíval jsem vodopádového modelu, kde byl kladen velký důraz na analýzu a návrh celého vizualizačního a informačního systému. V první řadě byl model uplatněn na problematiku ukládání dat do databáze, kde ve výsledku je zahrnuta funkce několika aplikací. Tato část pracuje samostatně, není nijak vázána na další vytvořené aplikace. Dále je má práce zaměřena na vizualizační a informační systémy, kde dvě aplikace přistupují k datům z databáze a nahlíží na aktuální data z OPC.

Výchozím bodem pro moji tvorbu se stalo OPC rozhraní, který z výsledku analýzy vyšlo nejlépe. Jako jediný standard je na obou měřicích systémech.

Další software, který přeposílá data z OPC standardu do databáze je zvolen DataHub, jehož funkce Data logging toto umožňuje. Je použita MS SQL databáze. Velký důraz byl kladen na optimální strukturu tabulek. Celý tento proces, kdy se přicházející data z OPC ukládají do databáze, bylo nutné navrhovat jako celek. Tohle ukládání funguje podle předpokladů déle než rok, během kterého byly odstraněny drobné nedostatky, které se objevily během provozu. Nyní můžu konstatovat, že systém běží stabilně a rychlost odezvy databáze na požadavek je velmi rychlá a bezporuchová. Přínosem pro mne byla nová zkušenost s databázemi a prakticky celá problematika okolo ukládání dat. Například použití SQL příkazů bylo velmi důležité pro následující přístup k informacím, které používají oba zrealizované vizualizační systémy.

Následující část, která prošla vlastním procesem návrhu a realizace je internetová a lokální vizualizace. Lokální vizualizace byla vytvořena ve SCADA systému Promotic. Podařilo se mi vytvořit potřebný náhled na data z databáze a z OPC. Problém se objevil při realizaci grafického zobrazování jednotlivých vrtů (při vyčítání dat z databáze). Předpoklad byl kladem na dva navigační prvky, které měly zobrazovat potřebné vrty ve formě tabulky a grafu současně, přičemž zvoleným požadavkem by došlo k překreslení obou objektů zároveň. Tento problém se mi podařilo vyřešit až samostatným přístupem, kde tabulka a graf je obsluhována odděleně.

V grafické vizualizaci jsou vytvořeny objekty, kde každý z nich je napojen na jednu tabulku v databázi. Ve vizualizaci je potom vytvořená konfigurace napojena k těmto objektům. Tím je zajištěno i prohlížení grafu. Tabulkový náhled je zajištěn přes předdefinované SQL dotazy.

Přínosem pro rozšíření mých znalostí, byla hlubší zkušenost v rámci realizace této lokální vizualizace.

Internetová vizualizace však plní všechny potřebné náležitosti, které jsem si stanovil za cíl. Při zvolení požadavku se zobrazí graf a dále tabulka. Požadavkem se rozumí v první řadě zvolení vrtu, poté začátek a konec časového rozmezí. Je tak možné zobrazení celého záznamu z databáze i zkráceného záznamu pouze s hodinovým výběrem. Data lze zobrazovat na webové stránce nebo je možné si je stáhnout v CSV souboru k vlastnímu zpracování, např. v Excelu. Při této realizaci jsem si osvojil práci v ASP.NET prostředí, kde jde hlavně o OOP a náležitostmi spojenými s webovými stránkami.

Při řešení celého systému byl zjištěn tento poznatek. Určitou nevýhodou může být ukládání surových dat v databázi. Aplikace, které přistupují k těmto datům, musí mít zajištěný převod, aby informace byly správně interpretovány. Například teplotní údaje z VVP jsou reálné i v databázi, takže nebyla nutná další úprava na straně kódu. Kdežto všechny tabulky z MVP bylo nutné přepočítat. To mělo za následek navrhování a přístup ke každé tabulce zvlášť. U MVP jsem vycházel z navrženého OPC serveru, kde byla zavedena myšlenka posílání dat v datovém typu INT (je nutný převod na desetinné místo pro správnou reprezentaci).

Struktura dat v databázi se osvědčila. K přehlednosti také přispělo navržení vždy jedné tabulky pro každý vrt.

Velkým přínosem je také ukládání dat v jeden časový okamžik. Data přibývají do každé tabulky se stejným časovým razítkem. Lze tedy snadno analyzovat vzájemné provázanosti mezi jednotlivými vrty. Např. vrt tepelného čerpadla a vrt vedlejší, který slouží k diagnostice, kde se teplotní změna projeví s časovým odstupem. Tuto skutečnost považuju za velmi přínosnou v rámci následné analýzy těchto dat.

Do budoucna je systém připraven na připojení dalšího měřicího systému (polygonu). Databázový systém, internetová i lokální vizualizace, které jsem vytvořil, je možné snadno rozšířit. Podklady k tomuto rozšíření se nachází v diplomové práci. Tento systém je plně otevřen i jiným možným variantám, například PHP. Do databáze může přistupovat každá internetová aplikace, která má možnost standardní komunikace s databázemi.

DataHub slouží nejen jako prostředník mezi OPC a databází. Bylo využito jeho další funkce, kdy kontroluje údaje z OPC, zda jsou v určitých mezích. Tyto reporty, pokud jsou překročeny, jsou posléze odeslány na email správci celého systému.

V rámci funkčnosti systému jsem provedl několik testů s kladným výsledkem. Jsou popsány všechny potřebné konfigurace a principy kódování, které pomohou k následujícímu rozšíření systému.

Literatura

- [1] Ošťádal, P.: *Vizualizační aplikace pro monitoring provozu tepelných čerpadel Nové auly VŠB-TU Ostrava*, Ostrava, Diplomová práce, 2009, Měřicí a řídicí technika
- [2] ICPDAS, *I-8KE4/I-8KE8* [online]. [cit. 2011-02-10].
Dostupné z WWW <http://www.icpdas.com/products/PAC/i-8000/i-8KE4_8KE8.htm>
- [3] Foxon, *Co je OPC?* [online]. [cit. 2011-02-14].
Dostupné z WWW <<http://www.foxon.cz/opc-server-opc-klient-c-72.html>>
- [4] Solid Quality Learning(organizace). *Microsoft SQL Server 2005 : základy databází krok za krokem*. Překlad Matějů, Veronika. Brno, Computer Press, a.s., 2007. ISBN 978-80-251-1524-4
- [5] Kofler, Michael. *Mistrovství v MySQL 5*. Překlad Svoboda, Jan. Baše, Ondřej. Černý, Jaroslav. Brno, Computer Press, a.s., 2007. ISBN 978-80-251-1502-2
- [6] AUTOMA časopis pro automatizační techniku, *Otevřený svět komunikací* [online]. [cit. 2011-03-20]. Dostupné z WWW
<http://www.odbornecasopisy.cz/index.php?id_document=28783>
- [7] Sensit, *Charakteristika čidla teploty PT 1000* [online]. [cit. 2011-03-21]. Dostupné z WWW
<http://www.sensit.cz/ke-stazeni/soubory/Pt1000_3850.pdf>
- [8] DataHub, *Chapter 6. Data Logging* [online]. [cit. 2011-03-29]. Dostupné z WWW
<<http://www.opcdatahub.com/Docs/dho-logging.html> >
- [9] DataHub, *Chapter 5. OPC to Email and SMS* [online]. [cit. 2011-04-01]. Dostupné z WWW
< <http://www.opcdatahub.com/Docs/dho-mailer.html> >
- [10] Evjen, Bill. *ASP.NET 2.0 : programujeme profesionálně*. Překlad Voráček, Karel. Brno, Computer Press, a.s., 2006. ISBN 80-251-1286-1

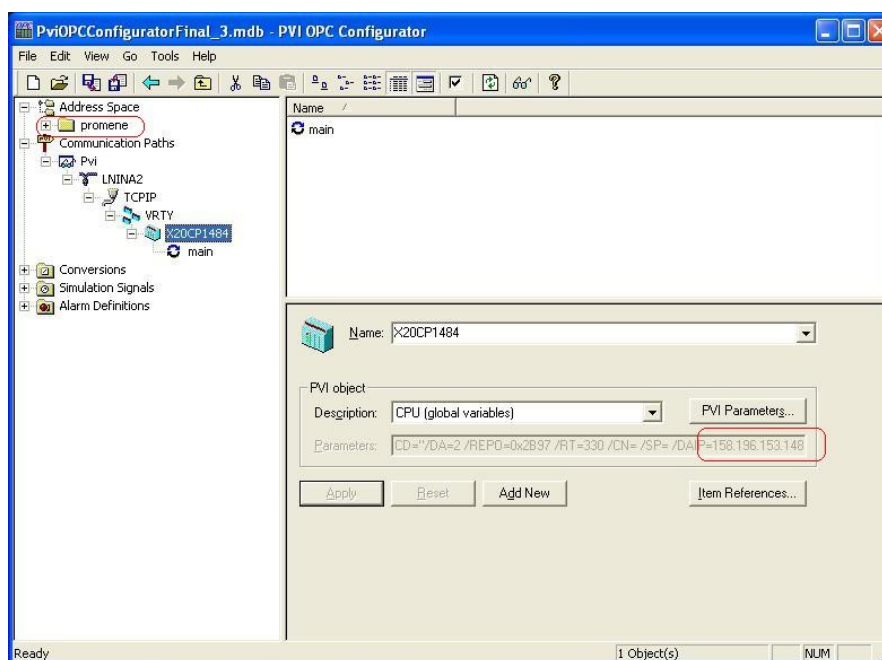
Seznam příloh

PŘÍLOHA I	-	B&R PviOPCConfigurator
PŘÍLOHA II	-	NAPOPC DA Server
PŘÍLOHA III	-	Nastavení ODBC propojení
PŘÍLOHA IV	-	Orientace v SQL Server Management Studiu
PŘÍLOHA V	-	Orientace v OPC DataHabu
PŘÍLOHA VI	-	OPCDataHub_AutoStart
PŘÍLOHA VII	-	Vizualizace Promotic
PŘÍLOHA VIII	-	OPC – Email
PŘÍLOHA IX	-	Vizualizace v Excelu (DDE)

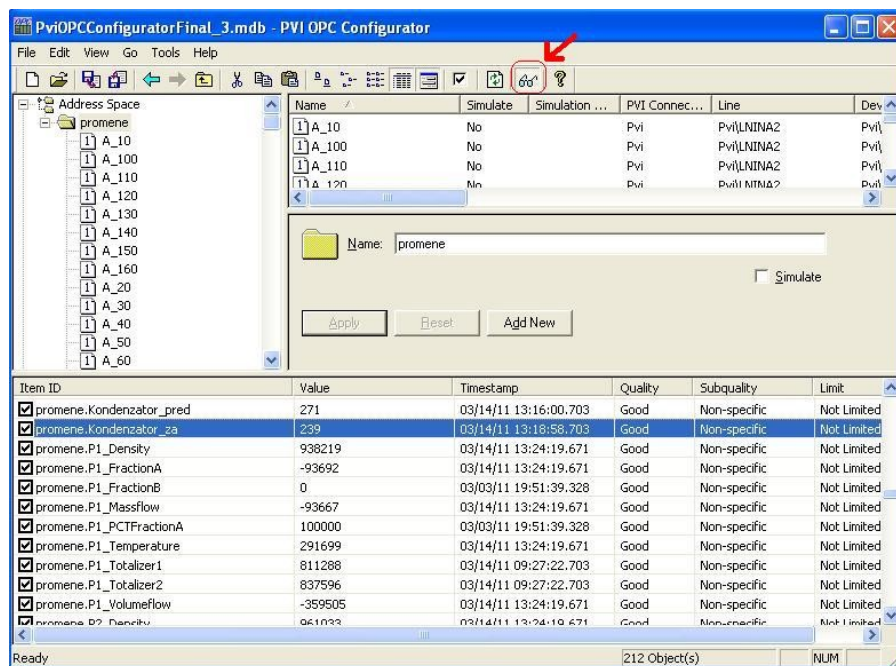
Příloha I – PviOPCConfigurator

PviOPCConfigurator slouží k nastavení přenášených hodnot přes OPC standard. Jde tedy o OPC server, který data dále poskytuje klientům. Níže na Obr. 1.1 je znázorněna IP adresa zařízení. Vlevo nahoře je adresář „proměnné“, který obsahuje dané data.

Obr. 1.1 Základní zobrazení OPC konfigurace

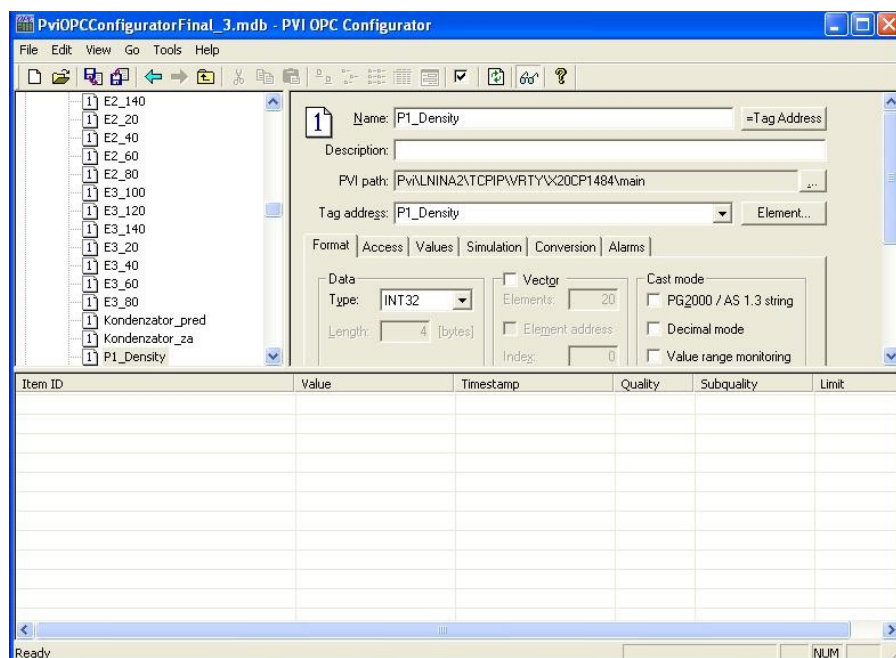


Obr. 1.2 Aktuální sledování OPC dat



Zkratka Alt-J zobrazí nastavení každé hodnoty. Na Obr. 1.3 je znázorněn Tag P1_Density. Lze vyčíst přiřazení Tag adresy k novému názvu, který je pojmenován stejně (pro přehlednost). Dále jde vidět datový typ, který je typu INT 32 a cesta v programu PLC (PVI path).

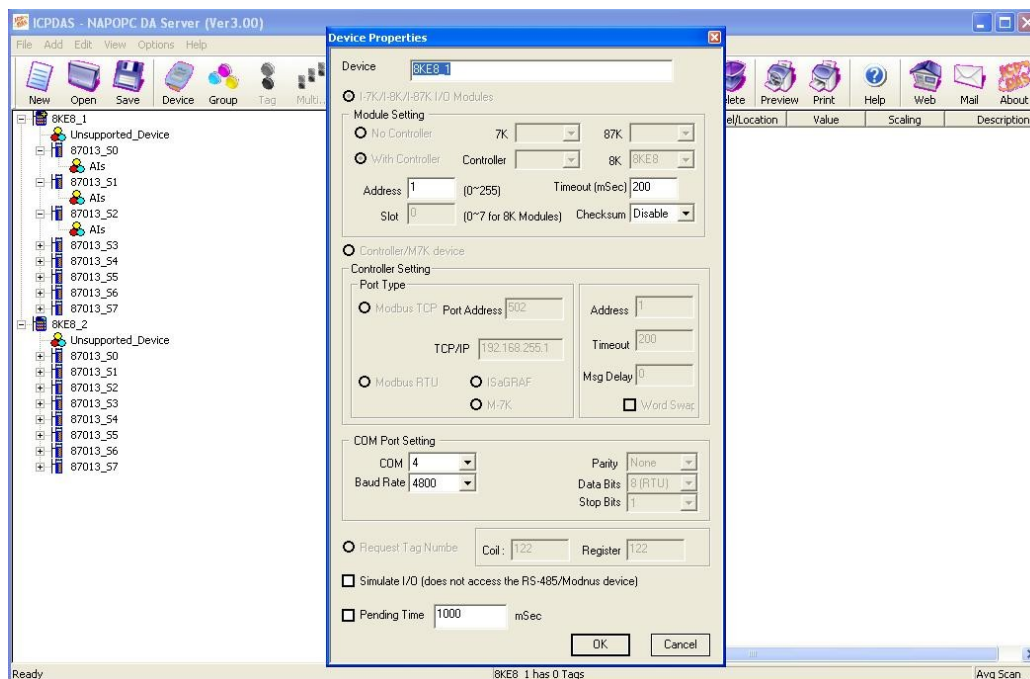
Obr. 1.3 Přidání proměnné



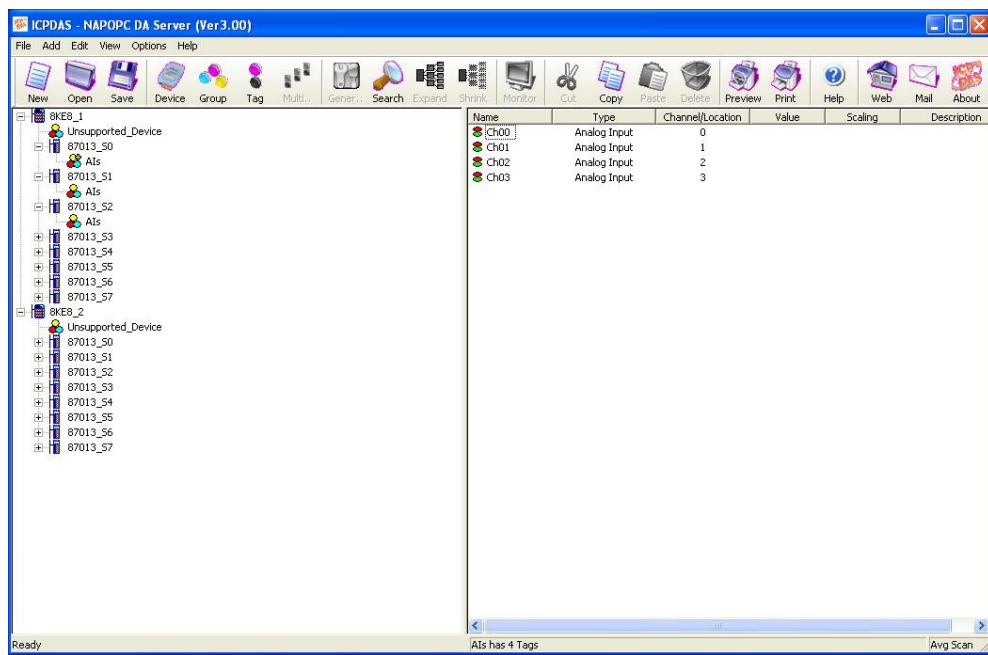
Příloha II – NAPOPC DA Server

Nadefinování OPC serveru ICPDASu vypadá následovně. Každá jednotka má definovanou rychlost a COM port. Jednodušší vyhledání jednotek je možné po stisknutí ikony „Search“. Nastaví se rychlost na 4800 Baud Rate a zvolí se COM port. Vyhledaná jednotka se zobrazí na levé straně okna. Proč se jednotky vyhledávají přes COM porty, bylo vysvětleno v kapitole 1.2.

Obr. 2.1 Nastavení ICPDASu



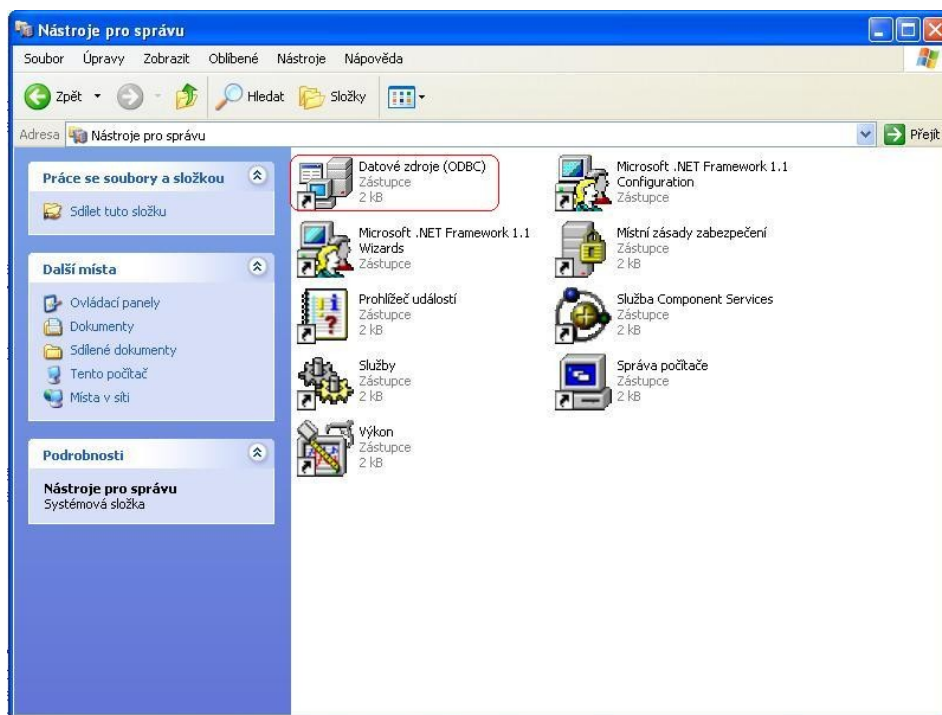
Obr. 2.2 Náhled na připojené jednotky



Příloha III – Nastavení ODBC propojení

ODBC datový zdroj (ODBC Data Source Administrator) je program, který je součástí Windows. Tento návod popisuje způsob propojení s databází (datovým zdrojem). Pro spuštění je nutné zvolit Ovládací panely – Nástroje pro správu – Datové zdroje (ODBC).

Obr. 3.1 ODBC datový zdroj

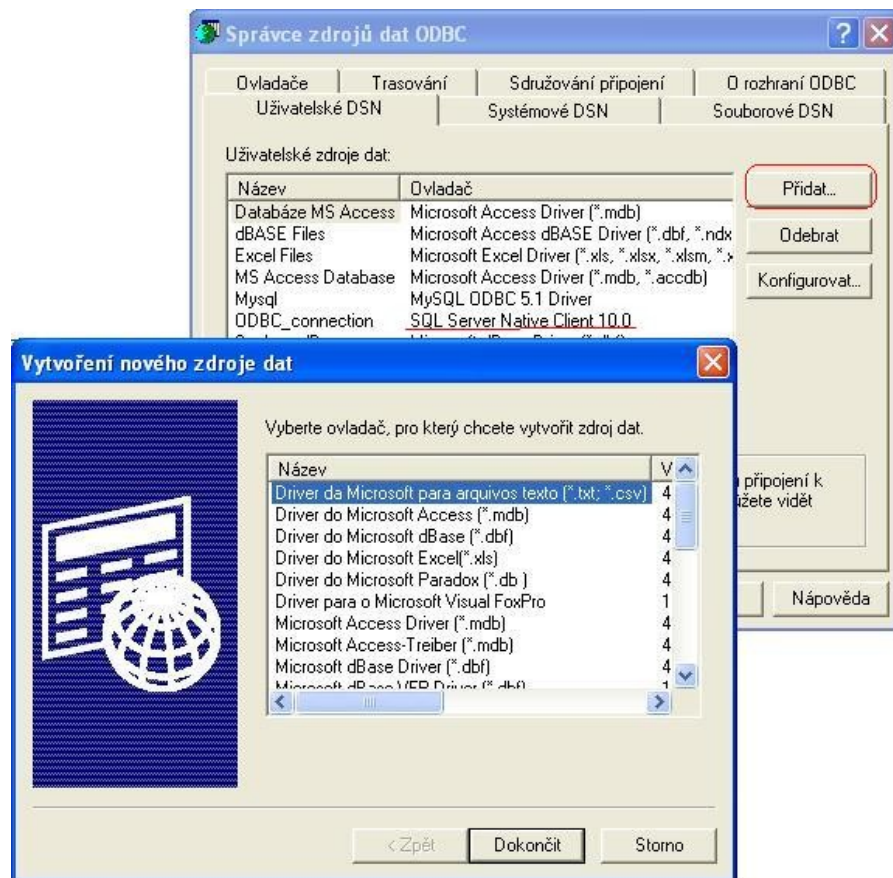


DSN (Data Source Name) znázorňuje způsoby propojení do databáze. Existuje Uživatelský DSN, Systémový DSN a Souborový DSN.

Byl vybrán Uživatelský DSN, ke kterému má přístup pouze přihlášený uživatel. Výhoda tkví v administrátorském účtu, který je krytý heslem. Žádný jiný uživatel nemůže propojení využít.

Zvolí se „Přidat“ a nalistuje se SQL Server Native Client 10.0, který byl vytvořen pro SQL Server 2008.

Obr. 3.2 Nový datový zdroj



Vyplní se:

- jméno „ODBC_connection“
- popisek
- vybere se název SQL Serveru 2008

Obr. 3.3 Nastavení DSN

Microsoft SQL Server DSN Configuration

This wizard will help you create an ODBC data source that you can use to connect to SQL Server.

What name do you want to use to refer to the data source?

Name: ODBC_connection

How do you want to describe the data source?

Description: datový zdroj

Which SQL Server do you want to connect to?

Server: VEC-VRTY\SQLEXPRESS

Dokončit Další > Storno Nápověda

- Heslo není použito
- Zaškrtně se připojení k SQL Serveru

Obr. 3.4 Další nastavení DSN

Microsoft SQL Server DSN Configuration

How should SQL Server verify the authenticity of the login ID?

☒ With Integrated Windows authentication.

SPN (Optional):

☐ With SQL Server authentication using a login ID and password entered by the user.

Login ID: bernecker

Password:

☒ Connect to SQL Server to obtain default settings for the additional configuration options.

< Zpět Další > Storno Nápověda

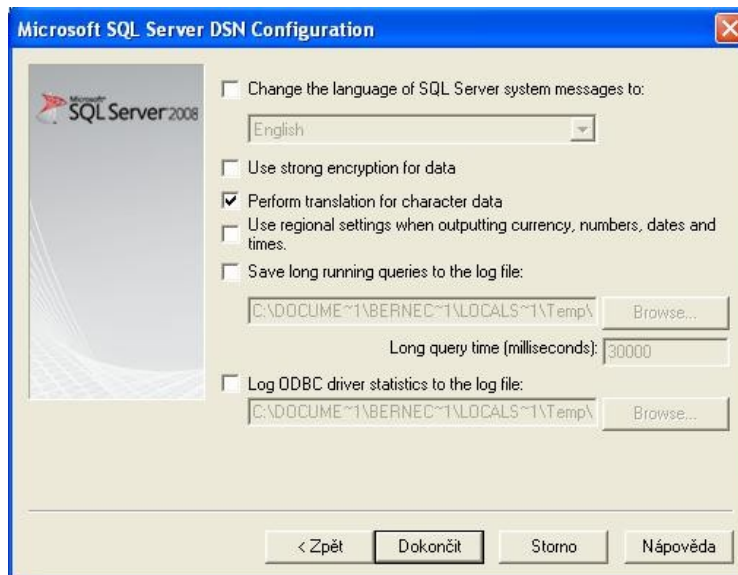
- Vybere se základní databáze „VRTY“

Obr. 3.5 Nastavení na výchozí tabulku v databázi



- Dále nastavení vypadá takto

Obr. 3.6 Dokončení nastavení



- Zvolí se dokončit a následně se provede ověření propojení

Příloha IV – Orientace v SQL Server Management Studiu

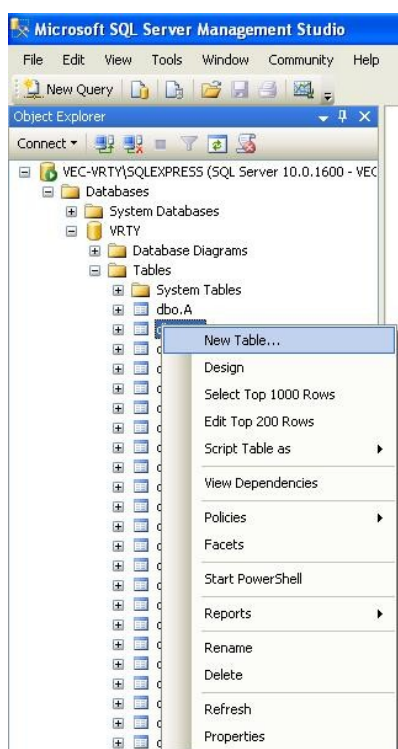
V nabídce programů se nachází složka Microsoft SQL Server 2008. Po spuštění SQL Server Management Studiu se zobrazí přístupové údaje. Hlavně „Server name“, kde je přesný název serveru. Autorizační přístup je řešen přes Windows Authentication.

Obr. 4.1 Připojení k SQL Serveru



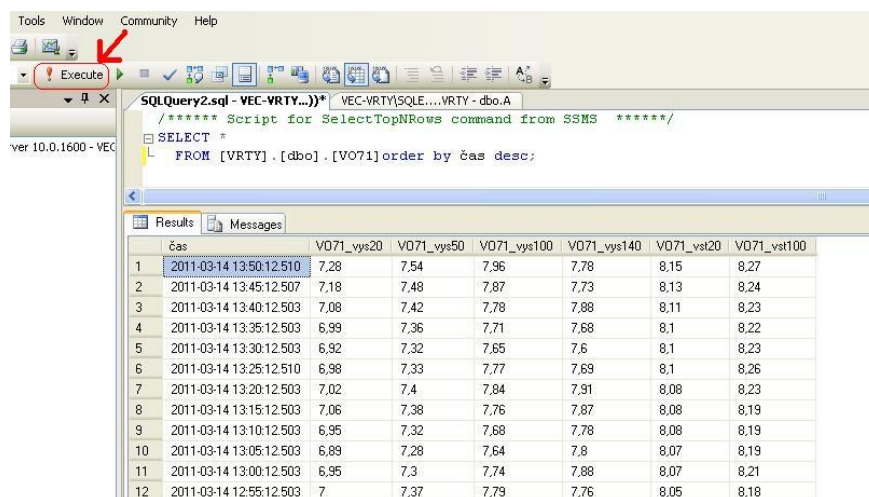
Po připojení k MS SQL Serveru se zobrazí pracovní prostředí. Po otevření záložky VRTY a záložky Tables se zobrazí vytvořené tabulky. Pravým tlačítkem myši se zobrazí okno s možnostmi vytvoření nové tabulky. Další je design, který zobrazí definici zvolené tabulky. Následující možnosti, např. Select Top 1000 Rows zobrazí 1000 záznamů ze zvolené tabulky. Jde o využití SQL příkazu, který tento požadavek zobrazí.

Obr. 4.2 Vytvoření nové tabulky



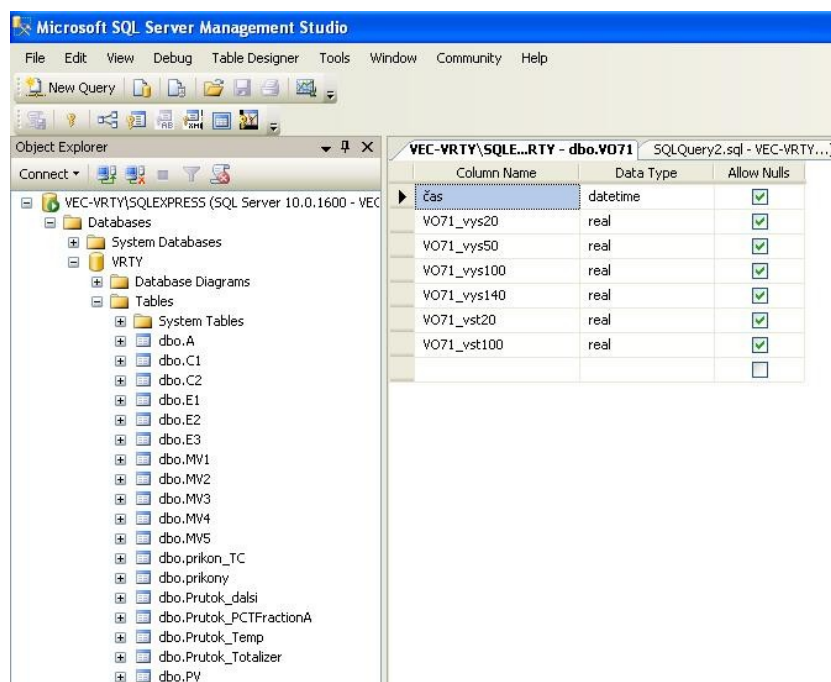
Je také možnost vytvoření vlastního příkazu. Následující Obr. 4.3 zobrazuje vytvořený dotaz pro SQL Server. Zobrazuje všechny záznamy z tabulky VO71, seřazené od nejnovějšího směrem dolů.

Obr. 4.3 Vytvoření SQL dotazu

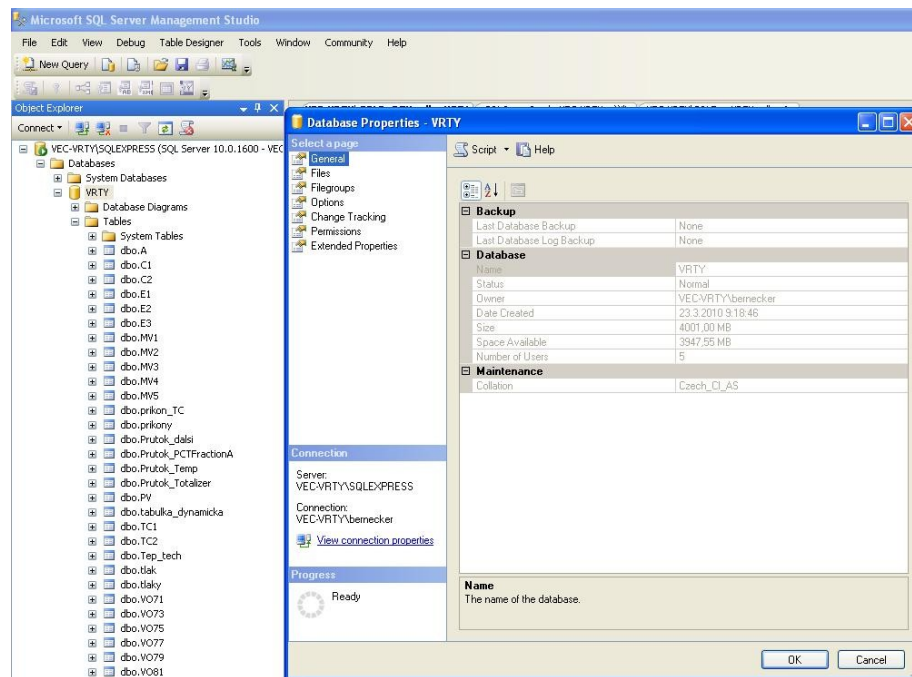


Na pravé straně obrázku je znázorněn Design vrtu VO71. Pro každou vytvořenou tabulku je třeba stejná definice datových typů. Takto definovaná tabulka je poté viditelná v DataHabu, kde se přiřazuje propojení s OPC servery, viz PŘÍLOHA V.

Obr. 4.4 Definice tabulky v databázi



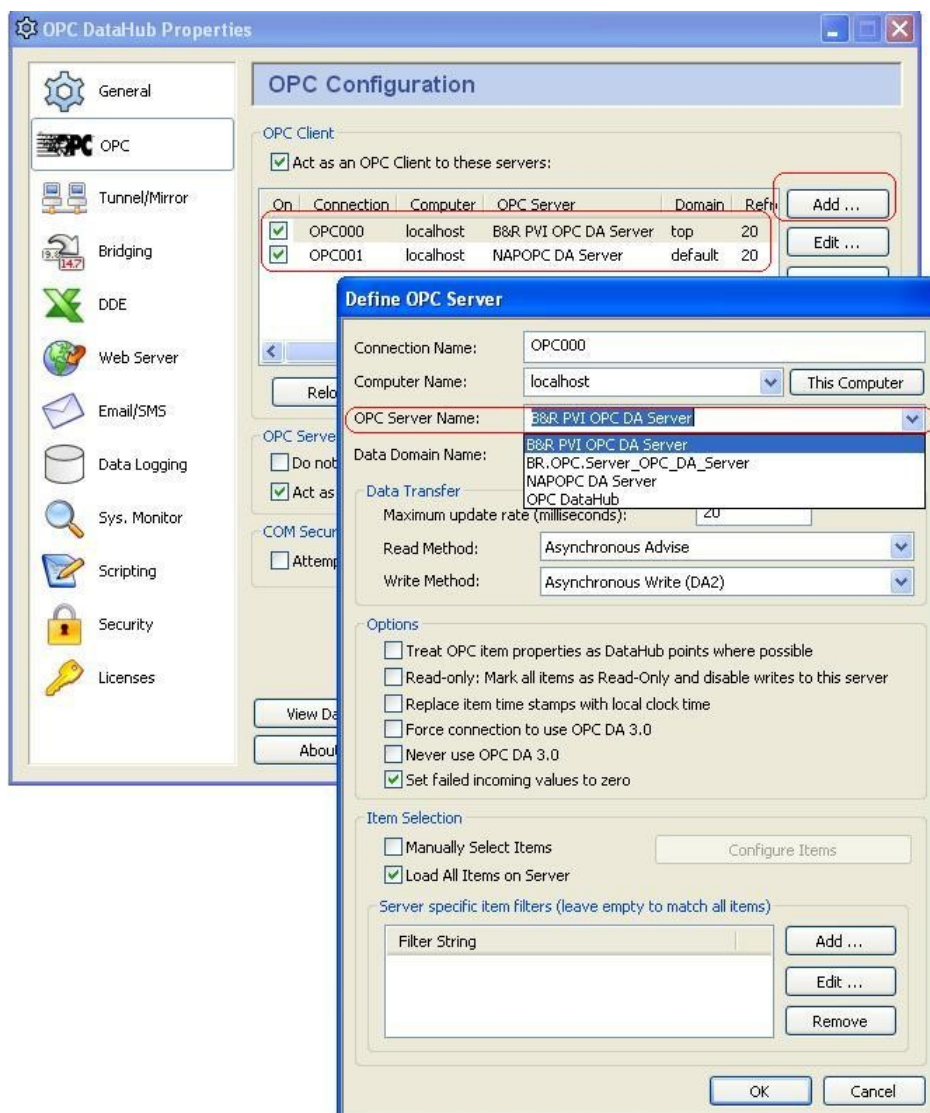
Obr. 4.5 Vlastnosti databáze



Příloha V – Orientace v OPC DataHabu

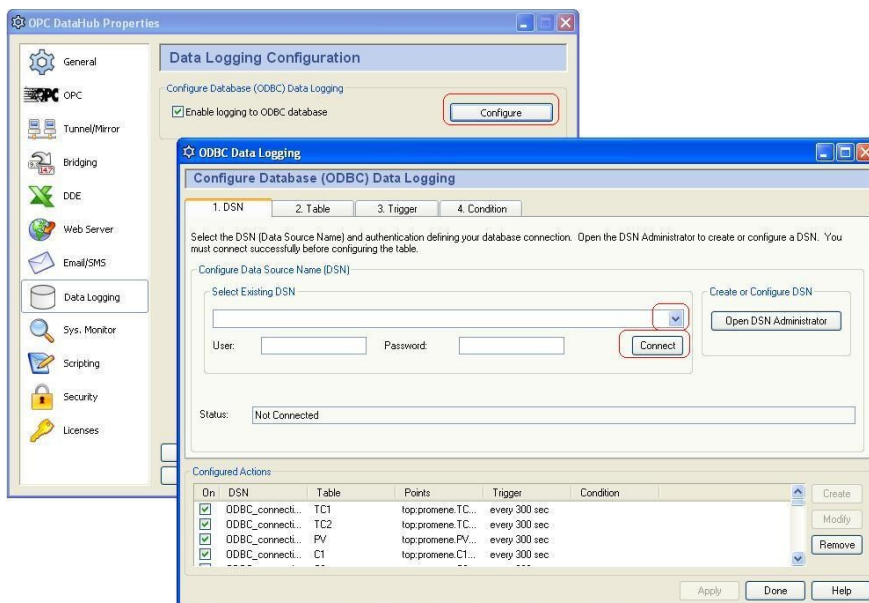
Tato problematika je nastudována z manuálu OPC DataHabu [8]. Níže je zobrazeno připojení k OPC serverům. V první řadě je nutné mít nainstalované potřebné OPC servery na PC. Poté v OPC DataHabu je možné tyto servery přiřadit v záložce „OPC“ tlačítkem „Add“. Po stisknutí se zobrazí další okno, kde se daný server vybere.

Obr. 5.1 Připojení OPC serveru



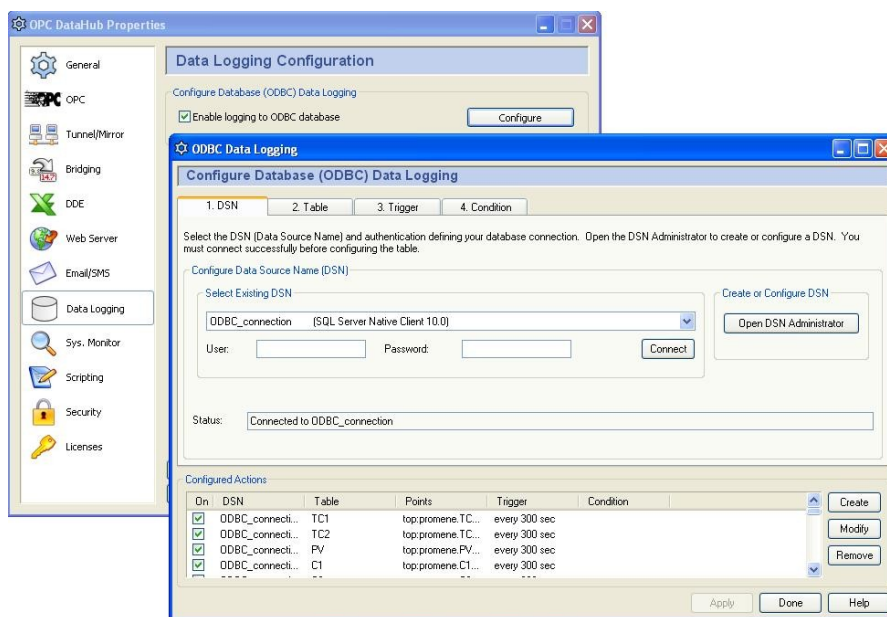
Následující obrázek zobrazuje Data Logging. Po stisku záložky Configure se zobrazí následující okno. Je nutné připojení k DSN datovému zdroji (přístup k databázi).

Obr. 5.2 Připojení k datovému zdroji



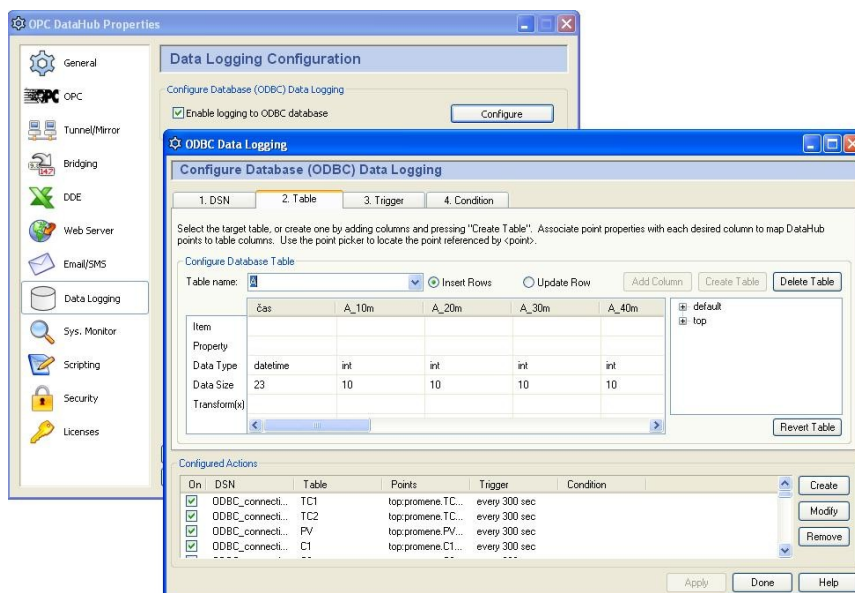
Datový zdroj, který byl vytvořen pro spojení s databází, se nazývá ODBC_connection. Stiskne se Connect.

Obr. 5.3 Připojený zdroj dat



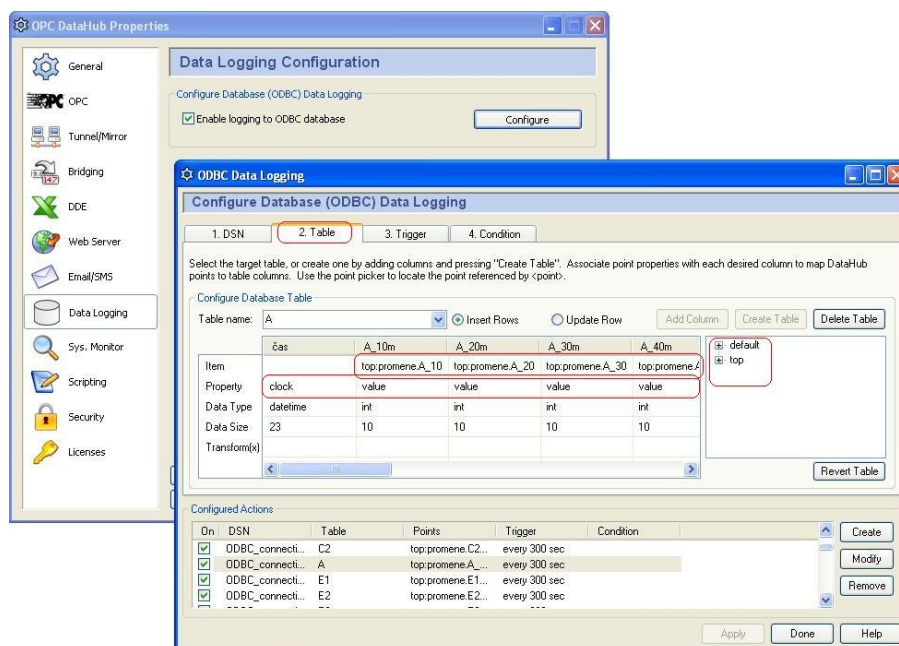
Po připojení k DSN, se přepne na další záložku „Table“. Jak je znázorněno na obrázku, jde vidět tabulka A a její definice datového typu pro každou kolonku. Tyto data jsou přebrané z databáze.

Obr. 5.4 Definování logování dat



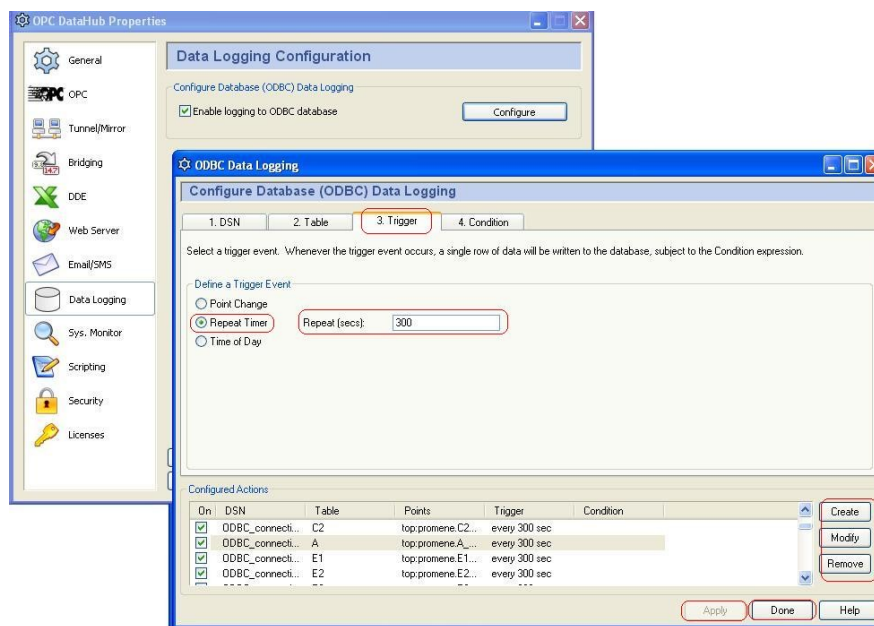
Prázdné hodnoty jsou vyplněny podle příslušných hodnot vybraných z MVP A VVP. Podle obrázku níže se jedná o tabulku A. Tabulka je součástí MVP, tyto data jsou v adresáři „top“. Data VVP jsou v adresáři „default“. Rozbalením adresáře top se zobrazí nabídka jednotlivých proměnných. Ty se pak přiřazují k jednotlivým kolonkám. Např. do kolonky A_10m se přiřadí hodnota „top:promene.A_10“. Jde tedy o intuitivní přiřazení, když se dodrží stejné pojmenování. Do kolonky „čas“ se vybere vlastnost „clock“. Jde o čas daný DataHubem (čas Windowsu).

Obr. 5.5 Definování propojení s databází



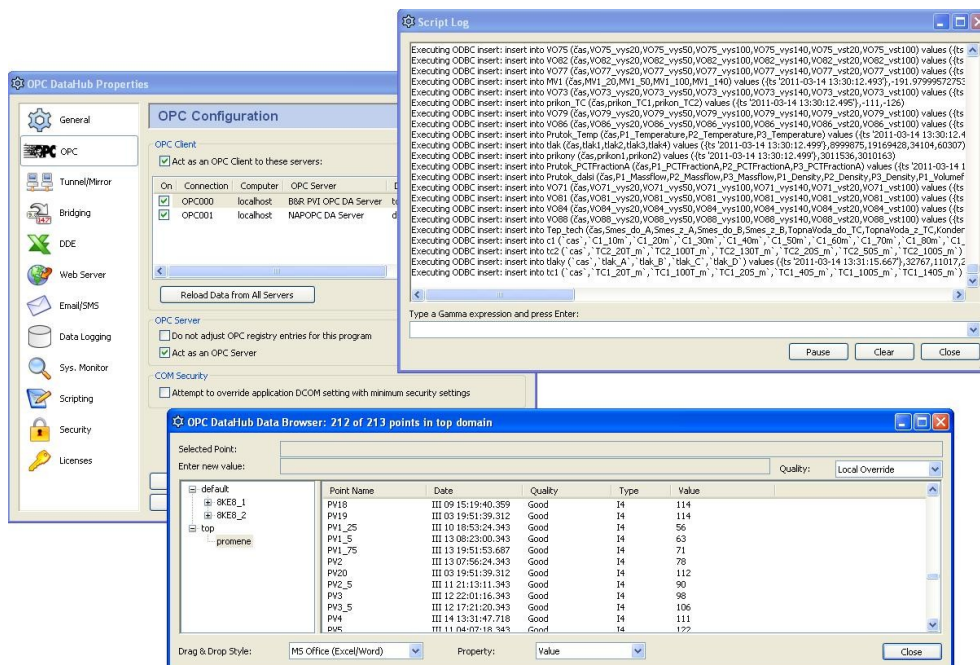
Záložka „Trigger“ odstartuje proces logování do databáze. Proto je zde nastaven čas opakování zápisu co 5 minut. Nyní je nastavení jedné tabulky dokončeno. Po stisku „Create“ se vytvoří konfigurace a zobrazí se v „Configured Actions“. Stiskne se „Apply“ a proces začne fungovat.

Obr. 5.6 Nastavení Triggeru



Následující obrázek zobrazuje Skriptovací okno, které znázorňuje proces logování. Lze také vidět případné chyby. Data Browser znázorňuje aktuální OPC data.

Obr. 5.7 Zobrazení Script Logu a Data Browseru



Příloha VI – OPCDataHub_AutoStart

Spuštění plné verze DataHubu verze 6. není možné jednoduchým kliknutím na ikonu DataHubu z důvodu české lokalizace Windows XP. Licenční soubor nemá přístup k potřebným souborům. Proto podle informací od vývojářů DataHubu je pro spuštění plné verze potřeba:

Vytvořit složku,

- c:\OPCDataHub\config

zkopírovat do ní konfiguraci z těchto složek, kde je nainstalován DataHub

- c:\documents and settings\bernecker\data aplikací\

a spuštění programu je třeba zapsat následovně.

- opcdatabus.exe -H c:\OPCDataHub\config

Po spuštění byl zapsán licenční soubor a DataHub pracuje v plném režimu.

Při vypnutí nebo restartu PC je vytvořen soubor nazvaný OPCDataHub_AutoStart.bat, který je uložen ve složce „Po spuštění“. Jeho spuštění se provede vždy po nastartování systému. Není třeba tuto zdlouhavou proceduru dělat manuálně. Jeho obsah vypadá následovně.

@echo on

cd c:\

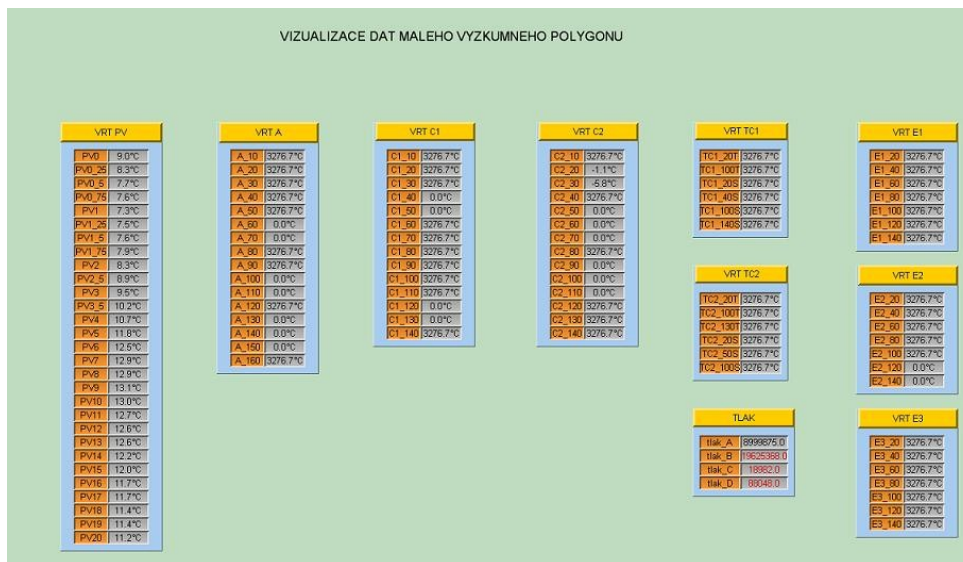
cd OPCDataHub\config\OPC DataHub

opcdatabus.exe -H c:\OPCDataHub\config

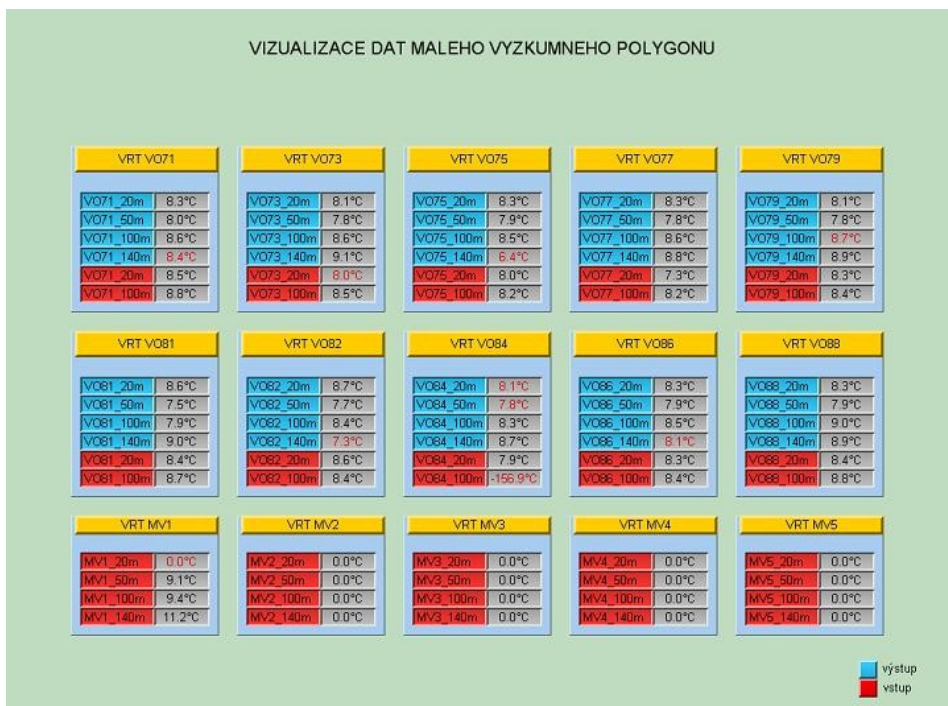
Příloha VII – Vizualizace Promotic

Znázornění zapnuté vizualizace MVP. Některé měřicí vrty byly vypnuty, z důvodu překopnutého kabelu.

Obr. 7.1 Vizualizace MVP



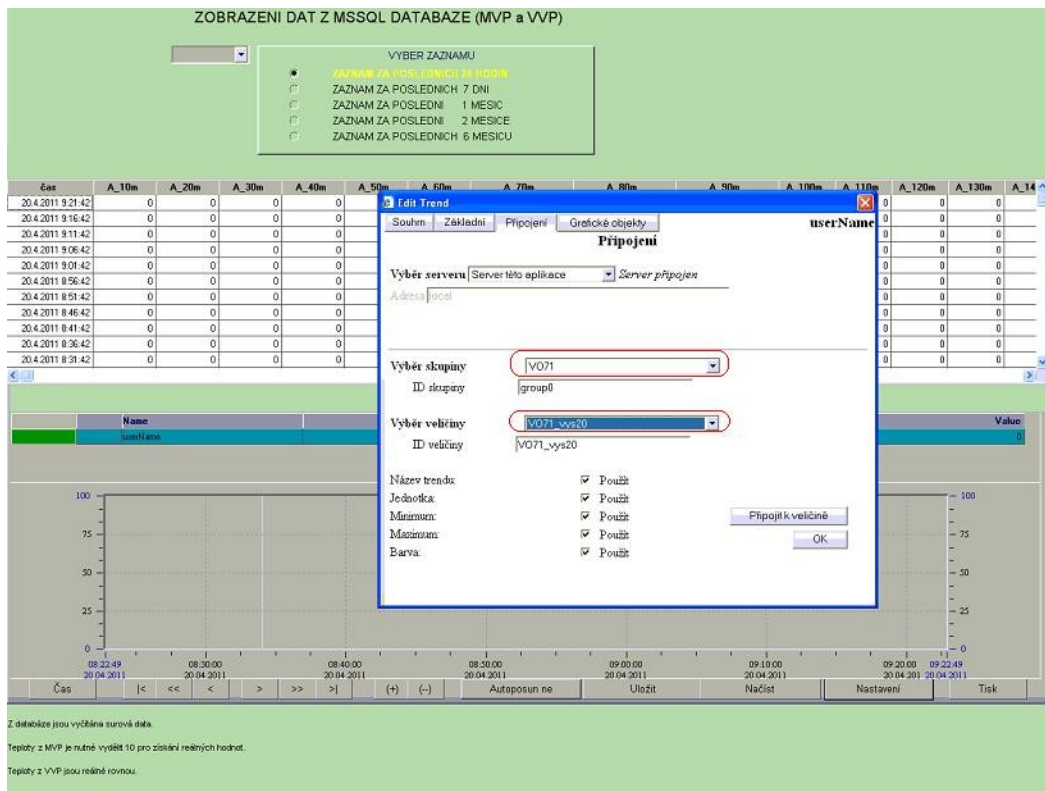
Obr. 7.2 Vizualizace VVP



U průběhu vizualizace VVP jsou červeně zobrazeny právě měnící se hodnoty.

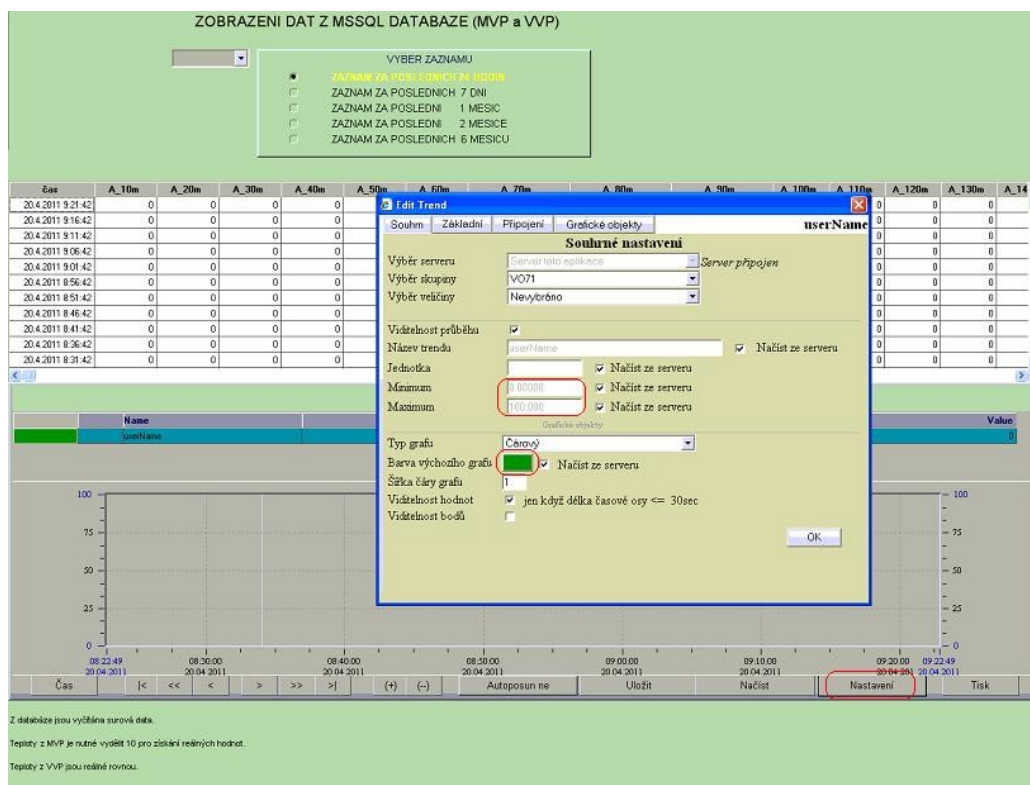
Po stisku tlačítka „Nastavení“ se zobrazí okno, kde je v záložce „Připojení“ možné vybrat některý z vrtů, které mají vytvořené fyzické propojení s databází přes objekt PmTrend. Dále se zvolí veličina a přejde se na záložku „Souhrn“, kde se nastaví zobrazované meze a barva průběhu, viz Obr. 7.4.

Obr. 7.3 Nastavení grafického zobrazení



Po nastavení maxima a minima se požadavek potvrdí a může se nadefinovat další.

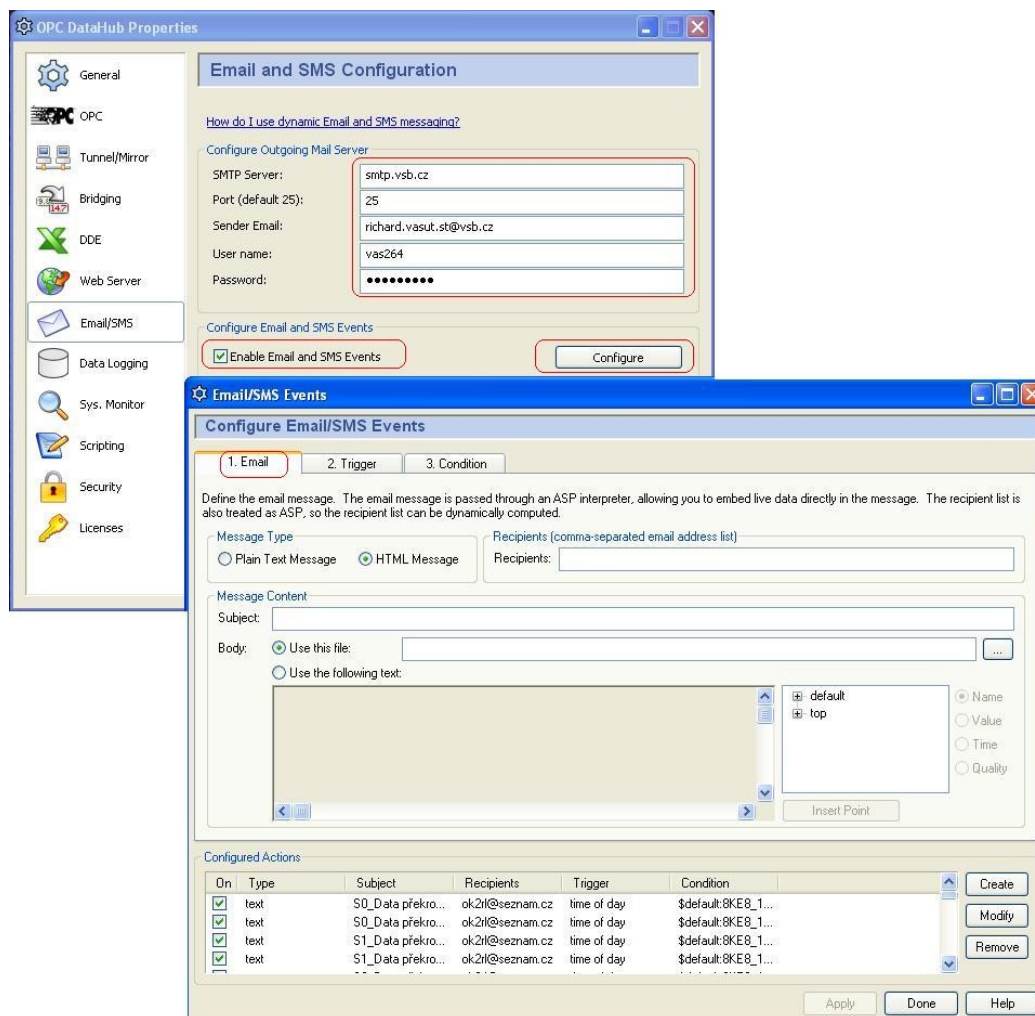
Obr. 7.4 Zvolení barvy a mezí



Příloha VIII – OPC – Email

Hlídaní mezi teplotních údajů je zajištěno DataHubem. V první řadě je nutné nastavení přístupu Dahanu k emailu. Na Obr. 8.1 je intuitivní náhled.

Obr 8.1 Nastavení Emailu



Konfigurace začíná v záložce „1. Email“, kde se nastaví příjemce zprávy společně s textem k odeslání. Na Obr. 8.2 jsou znázorněny čtyři proměnné, které se budou odesílat. V záložce Trigger se nastaví čas provedení, kde je nastaveno vykonání procesu čtyři dny v měsíci a to v čase 18:00:00. V záložce „Condition“ je možné maximálně nadefinovat čtyři podmínky. Proto i v obsahu zprávy jsou znázorněny ve stejném počtu. Zvolené proměnné se zkontrolují, zda jsou v daném rozsahu. Obr 8.4 znázorňuje, jestli údaje z vrtu A o hodnotách A_10 až A_40 nepřekročili hodnotu -300. Tato hodnota ve skutečnosti značí teplotu o velikosti

-30,0 °C. Stejná konfigurace je udělaná pro kladné hodnoty. Tím je zajištěno, že teploty, které se nebudou pohybovat v tomto rozmezí, budou nahlášeny správci systému.

Obr. 8.2 Nastavení textu a příjemce zprávy

Configure Email/SMS Events

1. Email 2. Trigger 3. Condition

Define the email message. The email message is passed through an ASP interpreter, allowing you to embed live data directly in the message. The recipient list is also treated as ASP, so the recipient list can be dynamically computed.

Message Type: ☒ Plain Text Message ☐ HTML Message

Recipients (comma-separated email address list): ok2tl@seznam.cz

Message Content

Subject: MVP_vrt_A_Data překročila povolený limit

Body: ☐ Use this file: D:\vasut_data\DataHub_HTML_text\varovani.txt ☒ Use the following text:

VRT A
.....
top.promene.A_10<%= \$top.promene.A_10 %>
top.promene.A_20<%= \$top.promene.A_20 %>
top.promene.A_30<%= \$top.promene.A_30 %>
top.promene.A_40<%= \$top.promene.A_40 %>
.....
konec

default
top

Insert Point

Configured Actions

On	Type	Subject	Recipients	Trigger	Condition
<input checked="" type="checkbox"/>	text	Jednotka_2_S6...	ok2tl@seznam.cz	time of day	\$default:8KE8_2...
<input checked="" type="checkbox"/>	text	Jednotka_2_S7...	ok2tl@seznam.cz	time of day	\$default:8KE8_2...
<input checked="" type="checkbox"/>	text	Jednotka_2_S7...	ok2tl@seznam.cz	time of day	\$default:8KE8_2...
<input checked="" type="checkbox"/>	text	MVP_vrt_A_Dat...	ok2tl@seznam.cz	time of day	\$top.promene.A...

Create Modify Remove Apply Done Help

Obr. 8.3 Provedení události jen v nastavené dny v čase 18:00

Configure Email/SMS Events

1. Email 2. Trigger 3. Condition

Select a trigger event. Whenever the trigger event occurs, the email/SMS message will be computed and transmitted, subject to the Condition expression.

Define a Trigger Event

☐ Point Change Year: * Month: * Hour: 18 Minute: 0 Second: 0

☒ Repeat Timer

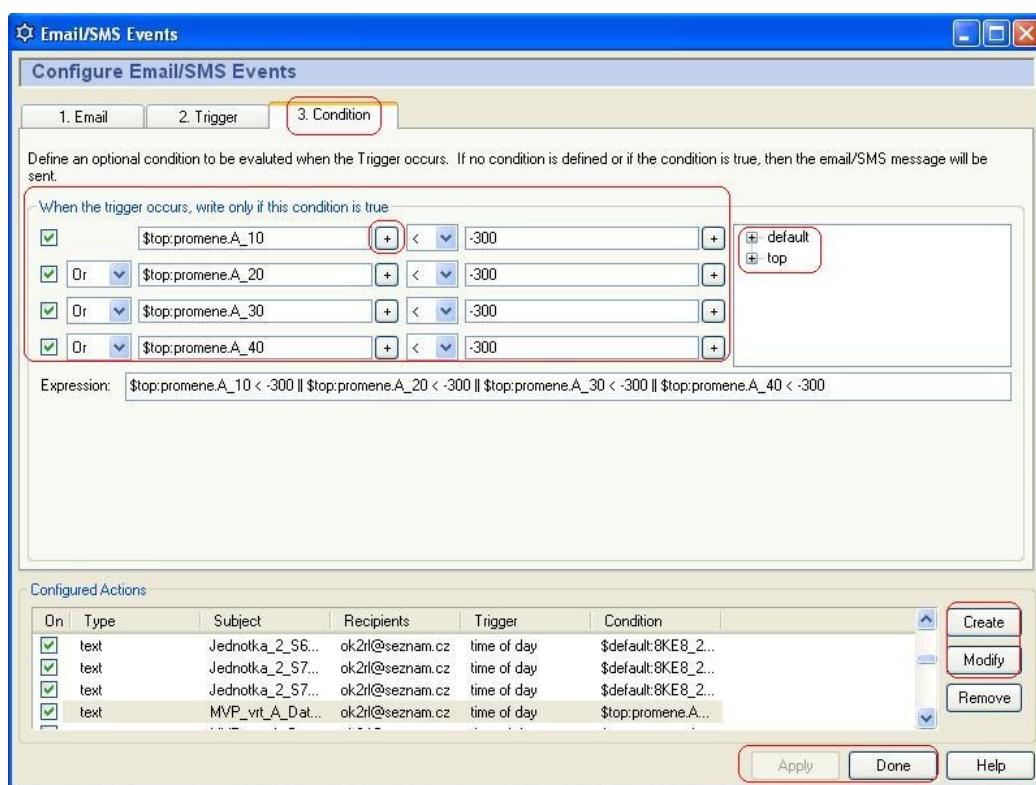
☒ Time of Day Day: 7,14,21,28

Configured Actions

On	Type	Subject	Recipients	Trigger	Condition
<input checked="" type="checkbox"/>	text	Jednotka_2_S6...	ok2tl@seznam.cz	time of day	\$default:8KE8_2...
<input checked="" type="checkbox"/>	text	Jednotka_2_S7...	ok2tl@seznam.cz	time of day	\$default:8KE8_2...
<input checked="" type="checkbox"/>	text	Jednotka_2_S7...	ok2tl@seznam.cz	time of day	\$default:8KE8_2...
<input checked="" type="checkbox"/>	text	MVP_vrt_A_Dat...	ok2tl@seznam.cz	time of day	\$top.promene.A...

Create Modify Remove Apply Done Help

Obr. 8.4 Nastavení podmínek



Příloha IX – Vizualizace v Excelu (DDE)

Obr. 9.1 Vizualizace v Excelu

Microsoft Excel - OPC DATA TABLE.xls																									
JIT																									
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
VVP																MVP									
	VO17 (%)				VO19 (%)				VO18 (%)				VO17 (%)				VO18 (%)				VO19 (%)				
1																									
2																									
3																									
4		20%	10.1	9.9	20%	9.9	9.9	20%	10.0	9.9	20%	10.1	9.9	20%	10.1	9.9	20%	10.1	9.9	20%	10.1	9.9	20%	10.1	9.9
5		30%	9.9	9.9	30%	9.4	9.4	30%	9.5	9.5	30%	9.3	30%	9.3	30%	9.3	30%	9.3	30%	9.3	30%	9.3	30%	9.3	30%
6		100%	9.9	9.9	100%	9.9	100%	9.9	100%	9.9	100%	10.0	100%	10.0	100%	10.0	100%	10.0	100%	10.0	100%	10.0	100%	10.0	100%
7		140%	10.5	10.5	140%	11.0	11.0	140%	9.9	9.9	140%	10.5	140%	10.5	140%	10.5	140%	10.5	140%	10.5	140%	10.5	140%	10.5	140%
8			10.2	9.7		9.7	9.7		9.7	9.7		9.6		9.6		10.1		10.1		10.1		10.1		10.1	
9			10.3	9.9		9.9	9.7		9.7	9.7		9.9		9.9		10.0		10.0		10.0		10.0		10.0	
10																									
11																									
12																									
13		20%	10.3	9.9	20%	10.0	9.8	20%	9.8	20%	9.9	20%	9.9	20%	10.1	9.9	20%	10.1	9.9	20%	10.1	9.9	20%	10.1	9.9
14		30%	9.1	9.9	30%	9.3	9.3	30%	9.3	30%	9.4	30%	9.4	30%	9.5	30%	9.5	30%	9.5	30%	9.5	30%	9.5	30%	9.5
15		100%	9.9	9.9	100%	9.9	100%	9.9	100%	9.6	100%	10.0	100%	10.0	100%	10.0	100%	10.0	100%	10.0	100%	10.0	100%	10.0	100%
16		140%	10.8	10.0	140%	9.4	10.0	140%	10.4	10.0	140%	10.0	140%	10.0	140%	10.9	140%	10.9	140%	10.9	140%	10.9	140%	10.9	140%
17			10.0	10.0		10.0	10.0		10.0	10.0		10.0		10.0		10.0		10.0		10.0		10.0		10.0	
18			10.3	9.9		10.0	9.7		9.9	9.7		10.0		10.0											
19		MV1 (%)				MV2 (%)				MV3 (%)				MV4 (%)				MV5 (%)							
20			0.0	0.0		0.0	0.0		0.0	0.0		0.0		0.0		0.0		0.0		0.0		0.0		0.0	
21			9.3	9.3		9.3	9.3		9.3	9.3		9.3		9.3		9.3		9.3		9.3		9.3		9.3	
22			9.3	9.3		9.3	9.3		9.3	9.3		9.3		9.3		9.3		9.3		9.3		9.3		9.3	
23			11.2	11.2		11.2	11.2		11.2	11.2		11.2		11.2		11.2		11.2		11.2		11.2		11.2	
24																									
25																									
26																									
27																									
28																									
29																									
30																									
31																									
32																									
33																									
34																									